

# Learning Video Object Segmentation from Limited Labelled Data

by

Mennatullah Siam

A thesis submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

Department of Computing Science

University of Alberta

© Mennatullah Siam, 2021

# Abstract

Video object/semantic segmentation has tremendous impact on many robotics applications. Videos of manipulation tasks or driving scenes are more relevant than static images. However, the focus of current video semantic segmentation work is on learning from large-scale datasets. Deep learning methods are highly data dependant, and require large amount of data to perform accurately. Manual annotation of large-scale video object/semantic segmentation benchmarks is labour intensive, and inefficient in terms of cost. Large companies and universities in first world countries have financed currently available benchmarks. The expense and massive computing needs and annotation cost creates a barrier to use deep learning with large-scale labelled data in e.g. developing countries. Thus, we focus on few-shot object segmentation which studies how to learn the segmentation of novel classes from few labelled sampled. Then we study its overlap with the video object segmentation task as a means to address the above problems. We present a thorough investigation of the shared challenges, assumption and solutions among both tasks.

Throughout the thesis contributions we mainly focus on metric learning approaches or what is also termed as learning to compare. We start with few-shot object segmentation and solve two main issues. The first issue we address is proposing a single branch method unlike previous methods that used two branches. We are inspired by cosine classifiers and propose a novel multi-resolution masked weight imprinting to generate the weights of the final segmentation layer for novel classes. The second issue we address is the use of

a single vector representation to guide the segmentation of novel classes which loses detailed information necessary for the segmentation task. We propose a co-attention mechanism with semantic conditioning to improve the interaction among the test (query set) and training (support set) data during few-shot inference. The semantic conditioning as well alleviates the need for pixel-level annotations for the few training data and rather depend on image-level labels.

We then transition to focus on video related tasks and formalize the task of video class agnostic segmentation that benefits from the overlap of few-shot and video object segmentation. We propose two formulations for the problem which focus on segmenting objects in a class agnostic manner and show applications in both autonomous driving and robot manipulation. The first formulation poses the problem as a motion segmentation problem, where we propose the first motion segmentation using deep learning in autonomous driving literature. We further provide KITTI-MoSeg dataset with motion segmentation annotations. Then we extend the work to incorporate motion instance labels along with increased number of categories to push the trained models to generalize to unknown moving objects. The second formulation as an open-set segmentation problem can handle both static and moving objects. We propose a novel contrastive learning approach with semantic and temporal guidance to improve the discrimination among known and unknown objects, and ensure temporal consistency. We further provide scenarios in the Carla simulation environment to motivate the reasons behind the need of such a formulation. Finally, we propose a motion adaptation mechanism for video class agnostic segmentation based on motion for an efficient inference.

*To My Mom*

*For always encouraging me and being by my side ♡.*

*The duty of the man who investigates the writings of scientists, if learning the truth is his goal, is to make himself an enemy of all that he reads, and attack it from every side. He should also suspect himself as he performs his critical examination of it, so that he may avoid falling into either prejudice or leniency.*

– AlHassan Ibn ElHaytham.

# Acknowledgements

I would like to thank my supervisor Professor Martin Jagersand for his support throughout the thesis and helping me to establish my basics in linear algebra and computer vision. I would also like to thank my supervising committee that provided guidance for me in the initial directions of my thesis. Some of the people that have helped me establish my thesis and I would not have been able to finish without his guidance and help in my work is Boris Oreshkin. He was a great mentor and collaborator that I have learned from throughout our work. I would also like to thank Senthil Yogamani for his help on my research work with Valeo vision systems. Finally, I want to thank and appreciate Alex Kendall for his support in my research work and mentorship.

The main people that I would have quit PhD without their support and love are my Mom, and my lovely three sisters whom I want to appreciate and thank for being there to me. I also want to thank my dear friends Noran elKafrawy, Mennatullah Hisham and Omnia Zayed for their support and kindness.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	2
1.2	Research Questions . . . . .	3
1.3	Thesis Contributions . . . . .	3
1.4	Authored Papers . . . . .	4
<b>2</b>	<b>Background and Related Work</b>	<b>6</b>
2.1	Semantic Segmentation . . . . .	6
2.1.1	Classical Methods . . . . .	7
2.1.2	Fully Convolutional Networks (FCN) . . . . .	9
2.1.3	Context Aware Models . . . . .	9
2.1.4	Video Semantic/Panoptic Segmentation . . . . .	11
2.2	Few-Shot Object Segmentation . . . . .	12
2.2.1	Few-shot Learning Setup and Taxonomy . . . . .	13
2.2.2	Problems Related to FSL . . . . .	13
2.2.3	Few-shot Object Segmentation Setup and Metrics . . . . .	16
2.2.4	Few-shot Object Segmentation Literature . . . . .	17
2.3	Video Object Segmentation . . . . .	19
2.3.1	Unsupervised Video Object Segmentation (UVOS) . . . . .	20
2.3.2	Semi-supervised Video Object Segmentation (SVOS) . . . . .	22
2.3.3	Motion Segmentation . . . . .	24
2.4	Intersection between FSS and VOS . . . . .	25
2.5	Background . . . . .	27
2.5.1	Transposed Convolution Arithmetics . . . . .	27
2.5.2	Learning Based approach for Optical Flow . . . . .	28
2.5.3	Self Supervised Learning of Depth and Pose . . . . .	29
<b>I</b>	<b>Few-shot Object Segmentation</b>	<b>31</b>
<b>3</b>	<b>Imprinting Masked Prototypes</b>	<b>32</b>
3.1	Introduction . . . . .	32
3.1.1	Metric Learning . . . . .	33
3.2	Proposed Method . . . . .	35
3.2.1	Few-shot Problem Setup . . . . .	35
3.2.2	Base Network . . . . .	36
3.2.3	Weight Imprinting . . . . .	36
3.2.4	Normalized Masked Average Pooling . . . . .	37
3.2.5	Adaptive Proxies . . . . .	38
3.2.6	Multiresolution Imprinting Scheme . . . . .	39
3.3	Experimental Results . . . . .	40
3.3.1	Experimental Setup . . . . .	41

3.3.2	1-Way Few-Shot Semantic Segmentation . . . . .	42
3.3.3	2-Way Few-Shot Semantic Segmentation . . . . .	44
3.3.4	Ablation Study . . . . .	45
3.4	Summary . . . . .	47
<b>4</b>	<b>Few-shot Weakly Supervised Object Segmentation using Co-Attention</b>	<b>48</b>
4.1	Introduction . . . . .	48
4.1.1	Attention Mechanisms . . . . .	51
4.2	Proposed Method . . . . .	52
4.2.1	Multi-Modal Interaction Module . . . . .	52
4.2.2	Stacked Gated Co-Attention . . . . .	54
4.2.3	Temporal Object Segmentation with a Few-shot Learning Setup . . . . .	55
4.3	Experimental Results . . . . .	56
4.3.1	Experimental Setup . . . . .	57
4.3.2	Comparison to the state-of-the-art . . . . .	59
4.3.3	Ablation Study . . . . .	60
4.4	Summary . . . . .	62
<b>II</b>	<b>Video Object Segmentation (VOS)</b>	<b>64</b>
<b>5</b>	<b>Video Class Agnostic Segmentation</b>	<b>65</b>
5.1	Introduction . . . . .	65
5.1.1	Unknown Objects segmentation . . . . .	68
5.1.2	Contrastive Learning . . . . .	69
5.2	Datasets . . . . .	69
5.2.1	Valeo KITTIMoSeg Dataset . . . . .	69
5.2.2	Wayve Datasets . . . . .	71
5.3	Motion Segmentation . . . . .	75
5.3.1	End-to-End Video Class Agnostic Segmentation . . . . .	75
5.3.2	Real-time Video Class Agnostic Segmentation . . . . .	76
5.3.3	Real-time Panoptic and Class Agnostic Segmentation . . . . .	78
5.4	Open-Set Segmentation . . . . .	82
5.4.1	Analysis on Unknown Objects . . . . .	83
5.4.2	Mahalanobis Based Segmentation . . . . .	84
5.4.3	Contrastive Learning with Semantic and Temporal Guidance . . . . .	85
5.5	Experimental Results . . . . .	87
5.5.1	Experimental Setup . . . . .	87
5.5.2	Datasets . . . . .	88
5.5.3	Evaluation Metrics . . . . .	89
5.5.4	Results and Discussion . . . . .	89
5.6	Summary . . . . .	101
<b>6</b>	<b>Motion Adaptation for Video Object Segmentation</b>	<b>102</b>
6.1	Introduction . . . . .	102
6.2	(I)nteractive (V)ideo (O)bject (S)egmentation (IVOS) Dataset	106
6.2.1	Human Teaching Objects . . . . .	107
6.2.2	Manipulation Tasks Setting . . . . .	107
6.3	Motion Adaptation . . . . .	108
6.3.1	Baseline Network Architecture . . . . .	108



6.3.2	Motion Adaptation using Pseudo-labels . . . . .	109
6.4	Experimental Results . . . . .	111
6.4.1	Experimental Setup . . . . .	111
6.4.2	Generic Video Object Segmentation . . . . .	112
6.4.3	Video Object Segmentation in HRI Setting . . . . .	114
6.5	Summary . . . . .	114
<b>7</b>	<b>Conclusion and Future Work</b>	<b>116</b>
7.1	Summary of Contributions . . . . .	116
7.2	Future Work . . . . .	118
7.3	Closing Remarks . . . . .	120
	<b>References</b>	<b>121</b>

# List of Tables

2.1	Few-Shot Learning Taxonomy. . . . .	14
2.2	Summary of Different Related Tasks to Few-shot Learning and their differences. Replica of Table from [48]. . . . .	15
2.3	Video Object Segmentation(VOS) Taxonomy. . . . .	20
2.4	Summary of FSS and VOS overlap. FSS: Fewshot segmentation. VOS: Video object segmentation. VCAS: video class agnostic segmentation. . . . .	26
3.1	Quantitative results for 1-way 1-shot and 5-shot segmentation on Pascal-5 <sup>i</sup> dataset following evaluation in [33]. FT: Fine-tuning for 10 iterations in 1-shot and 2 iterations in the 5-shot setting. . . . .	42
3.2	Quantitative results for 1-way 1-shot segmentation on Pascal-5 <sup>i</sup> dataset. FT: Fine-tuning. AMP-1: using Dilated FCN8s. AMP-2: using Reduced version of Dilated FCN8s. AMP-2 Norm: Similar to AMP-2 with using cosine similarity layer with normalization on features and weights during the training phase not only inference. Red, Blue: Best and Second Best Performing Methods. Co-FCN evaluation using Fg only reported from [203]. Bolded numbers indicate best performance. . . . .	43
3.3	Quantitative results for 1-way 5-shot segmentation on Pascal-5 <sup>i</sup> dataset. AMP-2 + FT(2): fine-tuning with 2 iterations after our proposed method. Red, Blue: Best and Second Best Performing Methods. Co-FCN evaluation using Fg only reported from [203]. Bolded numbers indicate best performance. . . . .	43
3.4	Quantative results on LFW [67] segmentation dataset. Rnd: Indicates the use of random weights in the final layer. MP: indicates imprinting the final layer weights using the masked proxies. FT: indicates fine-tuning. Bolded numbers indicate best performance. . . . .	45
3.5	Ablation study of the different design choices for the imprinting scheme on Fold 0. Adaptation: $\alpha$ parameter is non-zero. Multi-res: performing multi-resolution imprinting. Imp: imprinting weights using our proxies. FT: fine-tuning. Norm: Normalization. . . . .	46
3.6	Evaluating mean IoU over 15 base classes along with the extra novel class over 4 folds on Pascal-5i. Comparing our two variants based on the model trained with Cosine Similarity layer when fine-tuning with random weights or imprinting the weights and performing adaptation. . . . .	46
4.1	Notations Summary Table. . . . .	55

4.2	Quantitative results for 1-way, 1-shot segmentation on the Pascal-5 <sup>i</sup> dataset showing mean-Iou and binary-IoU. P: stands for using pixel-wise segmentation masks for supervision. IL: stands for using weak supervision from Image-Level labels. BB: stands for using bounding boxes for weak supervision. Red: validation following [201], Blue: validation following [177]. . . . .	59
4.3	Quantitative results for 1-way, 5-shot segmentation on the Pascal-5 <sup>i</sup> dataset showing mean-Iou and binary-IoU. P: stands for using pixel-wise segmentation masks for supervision. IL: stands for using weak supervision from Image-Level labels. BB: stands for using bounding boxes for weak supervision. Blue: validation following [177]. . . . .	60
4.4	Quantitative Results on MS-COCO Few-shot 1-way. . . . .	60
4.5	Ablation Study for different components with 1 run on Pascal-5 <sup>i</sup> and Youtube-VOS. V: visual, S: semantic. SCoAtt: Stack Co-Attention. Cond: Concatenation based conditioning. . . .	61
4.6	Ablation Study on 4 folds of Pascal-5 <sup>i</sup> for few-shot segmentation for different variants showing mean-IoU. V: visual, S: semantic. V+S: both features. . . . .	61
4.7	Quantitative Results on Youtube-VOS One-shot weakly supervised setup showing IoU per fold and mean-IoU over all folds similar to Pascal-5 <sup>i</sup> . V: visual, S: semantic. V+S: both features.	61
5.1	Comparison of different datasets for motion or primary object segmentation. Seqs: Sequences, Cats: Categories, Inst: Instances. Pan: Panoptic, Tr: Tracks. . . . .	71
5.2	Quantitative evaluation on KITTI MoSeg data for our proposed joint detection and motion segmentation network. . . . .	90
5.3	Comparison of different encoders and decoders on <b>Cityscapes validation set</b> . Evaluation is in terms of intersection over union and GFLOPs on image resolution 1024 × 512. Coarse indicates whether the network was pre-trained on the coarse annotation or not. . . . .	91
5.4	Comparison to the state of the art segmentation networks on <b>Cityscapes test set</b> in terms of intersection over union, and GFLOPs on image resolution <b>640x360</b> . . . . .	92
5.5	Quantitative results on KITTI-Motion in terms of mean intersection over union, mean average precision, running time and frame-rate on image resolution 384 × 768. . . . .	92
5.6	Results of CA and panoptic segmentation model. Tr-Te: Training and Test data used for CA segmentation. EFS: Ego Flow Suppression. D: DAVIS, K: KITTIMOTS Motion, C: Cityscapes-VPS Motion. Time is measured on image resolution 1024 × 2048 in seconds. . . . .	94
5.7	Open-Set Segmentation (VCAS-V2) Results on Cityscapes-VPS and CARLA. Fully supervised: Training the segmentation head on all cityscapes classes without a learnable global constant for the unknown object. CA-IoU: class agnostic IoU on the unknown objects. . . . .	95
5.8	Auxiliary loss factor for the Prototype and Temporal CL. . . .	98
5.9	Comparison between different Average Pooling Factors in Temporal CL. . . . .	98
5.10	Effect of Memory Queue in Prototype CL. . . . .	99

5.11	Quantitative Results on Cityscapes-VPS with larger batch size (4) and Step Learning Rate Scheduling. FD: Full Data. LU: Less number of objects labelled as unknown during training. . . . .	99
5.12	Quantitative Results on CARLA with less data and less number of objects labelled as unknown. LD: Less Data (2400 frames similar to Cityscapes-VPS). LU: number of pixels labelled as unknown during training. . . . .	101
6.1	Comparison of different datasets. T:Turntable, H:handheld. . . . .	105
6.2	Quantitative comparison on DAVIS'16 benchmark. MotAdapt-1: Continuous Labels, MotAdapt-2: Discrete Labels. . . . .	112
6.3	Quantitative results on FBMS dataset (test set). . . . .	112
6.4	mIoU on IVOS over the different transformations and tasks. IVOS dataset teaching is conducted on few samples from the translation, then evaluating on scale, rotation and manipulation tasks. MotAdapt-1: Continuous Labels. MotAdapt-2: Discrete Labels. . . . .	113

# List of Figures

2.1	Taxonomy of semantic segmentation approaches categorized into (1) Basic fully convolutional networks. (2) Models that incorporate contextual information. (3) Models that incorporate temporal information for video segmentation. . . . .	7
2.2	Two decoding methods for fully convolutional networks. The decoding method describes the approach for upsampling and computing the output pixel-wise labels. Figures adapted from [92], [130]. . . . .	10
2.3	Dilated (Atrous) Convolution with a 3x3 kernel with holes and Atrous Spatial Pyramid Pooling. Figures adapted from [18], [199].	11
2.4	OSLSM method for few-shot object segmentation. Figure from [136].	16
2.5	CoFCN method for few-shot object segmentation. Figure from [121].	17
2.6	CANet method for few-shot object segmentation. Figure from [201].	19
2.7	FusionSeg method Figure from [66]. . . . .	21
2.8	Coattention for Video Object Segmentation method Figure from [94].	23
2.9	(a) Convolution , (b) Transposed Convolution Operation as matrix multiplication, Figure from [103]. . . . .	27
2.10	Flow HSV Encoding, Figure from [5]. . . . .	29
2.11	Self Supervised Depth and Pose, Figure from [50]. (a) Depth Network. (b) Pose Network. (c) Photo-metric re-projection error, when handling occlusions the average loss will force occluded pixels to match, whereas the minimum loss only matches it to the visible.(d) Multi-scale. . . . .	30
3.1	Multi-resolution adaptive imprinting in AMP (Adaptive Masked Proxies). Imprinting occurs through masked average pooling of the final activations to be used as the novel class weights. The weights of the background class is further adapted. . . . .	33
3.2	Adaptive Masked Proxies using the Normalized Masked Average Pooling Layer. During few-shot inference two phases are computed: (1) Imprinting: which uses the support set image label pair. (2) Segmentation: which predicts the query image segmentation using the assigned weights from the first phase. For simplicity it shows the imprinting on the final layer solely. Nonetheless, our scheme is applied on multiple resolution levels.	35
3.3	Multi-resolution imprinting using proxies from different resolution levels. . . . .	40
3.4	Visualization for the T-SNE [96] embeddings for the generated masked proxies for novel classes unseen during training. Layers L1, L2, L3 denote the smaller to higher resolution feature maps.	41
3.5	Qualitative evaluation on Pascal-5 <sup>i</sup> 1-way 1-shot. The support set (a,c) and prediction (b,d) on the query image are shown. .	44

3.6	Output predictions on LFW dataset [67] using our proposed method without fine-tuning. (a,b) support set label-image pair. (c,d) query label-image pair. (e) our prediction. . . . .	45
4.1	Overview of stacked co-attention to relate the support set and query image using image-level labels. Nx: Co-attention stacked N times. “K-shot” refers to using K support images. . . . .	49
4.2	Architecture of Few-Shot Object segmentation model with co-attention. The $\oplus$ operator denotes concatenation, $\circ$ denotes element-wise multiplication. Only the decoder and multi-modal interaction module parameters are learned, while the encoder is pretrained on ImageNet. The stacked co-attention iterations are unrolled for visualisation solely. . . . .	53
4.3	Visual explanation of pixel-to-pixel affinity matrix used to compute attention weights. . . . .	54
4.4	Different variants for image-level labelled few-shot object segmentation. V+S: Stacked Co-Attention with Visual and Semantic representations. V: Co-Attention with Visual features only. S: Conditioning on semantic representation only from word embeddings. . . . .	56
4.5	Qualitative evaluation on Pascal-5 <sup>i</sup> 1-way 1-shot. The support set and prediction on the query image are shown in pairs. . . .	58
4.6	Failure cases on Pascal-5 <sup>i</sup> 1-way 1-shot. The support set and prediction on the query image are shown in pairs. . . . .	58
4.7	Qualitative analysis on fold 0 Pascal-5 <sup>i</sup> between our method (V+S) and object co-segmentation baseline ours (V) that can not disambiguate multiple common objects and is biased toward base classes used in training. . . . .	63
5.1	Video Class Agnostic Segmentation allows to identify objects outside the closed set of known classes. . . . .	66
5.2	Overview of the pipeline used to generate KITTI Moving Object Detection annotations. Blue boxes for moving vehicles, green boxes for static ones. . . . .	70
5.3	Curated Wayve motion dataset, extended annotations to real-world data from KITTI and Cityscapes. Red: Moving, Blue: Static. . . . .	72
5.4	Wayve Motion Dataset Statistics. . . . .	72
5.5	Different scenarios in CARLA Simulation and objects considered as unknown in our synthetic data. . . . .	73
5.6	Statistics for CARLA for both known classes and unknown objects, the different Towns in CARLA where our dataset was collected for training and testing phase and # Images. Blue: Known Classes, Red: Unknown objects. . . . .	74
5.7	MODNet Two Stream Multi-Task Learning Architecture for joint motion segmentation and object detection. Optical Flow and RGB input, RGB image with overlay motion segmentation in green and detected bounding boxes in blue. . . . .	76
5.8	Detailed Architecture of Two Stream ShuffleSeg for motion segmentation. SU: ShuffleNet Unit, #: denotes the number of channels. Modalities Fusion: fusion of appearance and motion features through concatenation of features. Feature Fusion: fusion of upsampled lower resolution feature maps and higher resolution maps. Blue: encoding layers, orange: decoding layers, yellow: modalities fusion. . . . .	78

5.9	Detailed Architecture for Class Agnostic and Panoptic Segmentation in Autonomous Driving. . . . .	80
5.10	Adopted from [179] Vanilla SOLO versus decoupled SOLO heads with detailed explanation for mask prediction branch. . . . .	81
5.11	(a) Computed output from ego-flow suppression. (b) Original flow. . . . .	82
5.12	Video Class Agnostic Segmentation using Contrastive Learning.	83
5.13	Dendrogram among unknown objects used during training and testing phases. . . . .	84
5.14	Qualitative evaluation on KITTIMoSeg data for our proposed two-stream multi-task learning network MODNet. top row: Input Optical Flow, middle row output of 2 tasks: overlay motion mask (green), bottom row output of 3 tasks: overlay motion mask (yellow), road segmentation (green) and detected bounding boxes (blue). . . . .	90
5.15	Precision-Recall Curve on KITTI-Motion. Our method provides on par results in detection (PR) with $4\times$ speedup. . . .	93
5.16	Top: predicted panoptic segmentation. Bottom: predicted CA segmentation on (a) KITTIMOTS Motion. (b) Cityscapes-VPS Motion. (c) IDD. . . . .	94
5.17	CA-IoU reported per scenario. Predicted semantic and class agnostic segmentation on CARLA Scenarios (a) Construction. (b) Barrier. (c) Parking. Top: semantic segmentation. Bottom: class agnostic segmentation. . . . .	96
5.18	Predicted semantic and class agnostic segmentation on Cityscapes-VPS. Top: semantic segmentation. Bottom: class agnostic segmentation (Note: pedestrian, rider, bicycle and motorcycle are withheld from training). . . . .	96
5.19	T-SNE visualisation of the masked embeddings for 15 known classes along with the unknown objects. (a) No Contrastive Learning. (b) Prototype-level Contrastive Learning. (c) Class statistics in Cityscapes-VPS. . . . .	97
6.1	Overview of the proposed Teacher-Student adaptation method for video object segmentation. The teacher model based on motion cues is able to provide pseudo-labels to adapt the student model. Blue: confident positive pixels. Cyan: ignored region in the adaptation. . . . .	103
6.2	Samples of collected Dataset IVOS, Teaching Objects Setting. (a) Translation split. (b) Scale split. (c) Planar Rotation split. (d) Out-of-plane Rotation. . . . .	106
6.3	Samples of collected IVOS dataset, Robot manipulation Tasks Setting with segmentation annotation. Manipulation Tasks: (a) Drinking. (b) Stirring. (c) Pouring Water. (d) Pouring Cereal. . . . .	107
6.4	Motion Adaptation of fully convolutional residual networks pipeline.	108
6.5	(a,b) Discrete adaptation targets (pseudo-labels), cyan is the unknown region, blue is the confident positive pixels. (c, d) Continuous adaptation targets. . . . .	111
6.6	Qualitative Evaluation on the FBMS dataset. Top: LVO [165]. Bottom: ours. . . . .	112
6.7	Qualitative evaluation on DAVIS'16. (a) LVO [165]. (b) ARP [77]. (c) Baseline. (d) MotAdapt. . . . .	113

6.8 Qualitative evaluation on IVOS Manipulation Tasks Setting.  
(a) Teaching Phase, Discrete Labels. (b) Teaching Phase, Continuous Labels. (c) Inference Phase before manipulation. (d) Inference Phase, during manipulation. . . . . 113



# Chapter 1

## Introduction

Semantic segmentation predicts pixel-wise labels for the scene which provides a means for scene understanding necessary in various robotics applications such as autonomous driving [25][131] and robot manipulation [32][70]. While there has been recent success in deep learning approaches for semantic segmentation and video object segmentation (e.g., [18][20][165][66][166]), current approaches depend mainly on prior large-scale training data [87][40]. However, relying solely on a closed set of known and limited objects that exist in large-scale manually labelled datasets limits the applicability of these deep networks in real world problems [30][143][117]. This thesis focuses on learning **class agnostic segmentation** i.e. to segment objects relying on appearance, motion and/or depth cues in a class agnostic manner in order to handle unknown objects outside the closed set of known classes. The class agnostic segmentation problem was initially defined in the **few-shot object segmentation** literature which learns to segment images based on few pixel-level labelled support set. We rather focus on its extension to video sequences from monocular cameras through utilizing motion and depth cues. The **video class agnostic segmentation** problem has relation to **zero/few-shot video object segmentation**. Video object segmentation is concerned with either segmenting the visually salient object in a video sequence (zero-shot) or propagating the initial segmentation masks from few labelled frames (few-shot).

## 1.1 Motivation

Semantic segmentation using deep learning generally requires large-scale annotated datasets such as Cityscapes [25], Mapillary [104], Synthia [131]. Annotations are rather labour intensive and time inefficient, it is even worse for video object segmentation annotations since it handles video sequences not just frames. Large-scale semantic segmentation datasets such as MS-COCO [87] has annotations for only 80 objects, the largest dataset up to date is Open-Images [6] which has around 600 classes. As indicated in [184] the number of English nouns range between 500,000 to 700,000 which suggests the enormous variability in objects that can be encountered. Robotics environments are unstructured and dynamic, and could contain a different variety of object classes. Adapting to that ever-changing environment is a challenging endeavour. Learning from limited labelled data has a great potential to overcome such difficulty and gets us closer to human like intelligence. Humans, especially children, have the ability to generalize to new classes with few labelled samples [97] or none at all based on motion, depth cues or through interacting with that object. Thus, it is of great interest for both the computer vision and robotics community to mimic human like intelligence in that aspect.

Learning from limited labelled data has its benefits as well in decolonizing artificial intelligence and directing it to become more inclusive. In an article by a research scientist in Deepmind AI [29], it was mentioned that New York Times article [82] discussing the future with AI for the different countries explicitly stated: “Unless they wish to plunge their people into poverty, they will be forced to negotiate with whichever country supplies most of their A.I. software — China or the United States — to essentially become that country’s economic dependent.” That quote demonstrates the major problems from an exclusive AI that is dominated by large-scale corporations. It is currently considered that “Data is the new oil”, it highly indicates that developing nations that do not have the same resources to collect and annotate data as developed ones will suffer greatly. This motivates the focus of the thesis on learning few-shot object segmentation where usually one to five samples are provided

as the training data for novel classes. It also relates to few-shot and zero-shot video object segmentation. In zero-shot video object segmentation the model is expected to segment the primary object through a video sequence in a class agnostic manner. While in few-shot video object segmentation the model is provided with a single labelled frame that acts as a training instance to learn to segment that object in the video frames. Both video object segmentation methods can benefit from motion cues, that can act as a strong indicator of the object location or can be used to relate the object instances in consecutive frames.

## 1.2 Research Questions

In our work we mainly focus on four research questions:

- In few-shot object segmentation how to segment a query image with novel classes using a single branched method (single set of network parameters)?
- In few-shot object segmentation how to leverage the interaction between query and support set especially with image-level labels only?
- How to learn video object segmentation from limited or no labelled data?
- How to segment pixels outside of the closed set of known classes? What cues to utilize for such a task? How autonomous driving can benefit from such a task?

## 1.3 Thesis Contributions

The contributions of the thesis are as follows:

- The main contribution is introducing the task of video class agnostic segmentation where the goal is to segment instances of unknown objects through utilizing appearance, motion and geometry. It is mainly inspired by few-shot learning and specifically metric learning.

- We further propose a setup for temporal few-shot learning and show different ways to investigate the overlap between video object segmentation and few-shot object segmentation.
- We demonstrate how to leverage few-shot object segmentation methods using metric learning, and attention mechanisms to achieve the goal of generalizing from base classes with large-scale data to novel classes with few labelled samples.
- We innovate methods for zero-shot video object segmentation that utilize motion representation in terms of optical flow.

## 1.4 Authored Papers

Some extracts from this thesis appear in the following authored publications and preprints.

Part I contains work on few-shot object segmentation from:

- Siam, Mennatullah, Boris N. Oreshkin, and Martin Jagersand. “AMP: Adaptive Masked Proxies for Few-Shot Segmentation.” Proceedings of the IEEE International Conference on Computer Vision. 2019.
- Siam, Mennatullah et al. “Weakly Supervised Few-shot Object Segmentation using Co-Attention with Visual and Semantic Embeddings.” Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20. International Joint Conferences on Artificial Intelligence Organization.

Part II contains work on video object segmentation from:

- Siam, Mennatullah, et al. “Video object segmentation using teacher-student adaptation in a human robot interaction (HRI) setting.” 2019 International Conference on Robotics and Automation (ICRA). IEEE, 2019.

- Siam, Mennatullah, et al. “Real-Time Segmentation with Appearance, Motion and Geometry.” 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.
- Siam, Mennatullah, et al. “Modnet: Motion and appearance based moving object detection network for autonomous driving.” 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2018.
- Siam, Mennatullah, Alex Kendall, and Martin Jagersand. “VCA: Video Class Agnostic Segmentation with Contrastive Learning in Autonomous Driving.” Under Review for IEEE International Conference on Computer Vision. 2021.

A conducted survey on deep semantic segmentation and video object segmentation for automated driving was published during the thesis work as well:

- Siam, Mennatullah, et al. “Deep semantic segmentation for automated driving: Taxonomy, roadmap and challenges.” 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC). IEEE, 2017.

A patent on moving object segmentation in a class agnostic manner was published during the thesis work as well with Valeo Vision Systems:

- Siam, Mennatullah; Yogamani, Senthil Kumar; El-Sallab, Ahmad; Mahgoub, Heba “Verfahren zum Bestimmen eines Bewegungszustands eines Objekts in Abhängigkeit einer erzeugten Bewegungsmaske und eines erzeugten Begrenzungsrahmens, Fahrerassistenzsystem sowie Kraftfahrzeug”.<sup>1</sup>

---

<sup>1</sup>[https://worldwide.espacenet.com/publicationDetails/biblio?CC=DE&NR=102018114229&KC=&FT=E&locale=en\\_EP#](https://worldwide.espacenet.com/publicationDetails/biblio?CC=DE&NR=102018114229&KC=&FT=E&locale=en_EP#)

# Chapter 2

## Background and Related Work

We mainly focus on the two concurrent and overlapping topics few-shot object segmentation and video object segmentation. We first introduce classical and deep semantic segmentation literature in Section 2.1, then we cover the few-shot and video object segmentation literature in Sections 2.2 and 2.3 respectively. Then section 2.4 discusses the overlapping research problems between both areas and how to benefit each other. Finally we introduce the necessary background in Section 2.5.

### 2.1 Semantic Segmentation

Semantic segmentation has long been studied in classical methods before the emergence of deep learning, but was always bounded by the performance of classifiers relying on hand crafted features. After the emergence of deep convolutional neural networks, semantic segmentation has shown greater promise for deployment in real-world applications, with some methods inspiring from classical graph based approaches. The literature in deep semantic segmentation is categorized into three groups: **(1)** Basic Fully Convolutional Networks. **(2)** Context Aware Models. **(2)** Temporal Models for video segmentation. Figure 2.1 summarizes the segmentation literature within these three categories. In the following sections we cover the classical methods, and the three main categories in deep semantic segmentation.

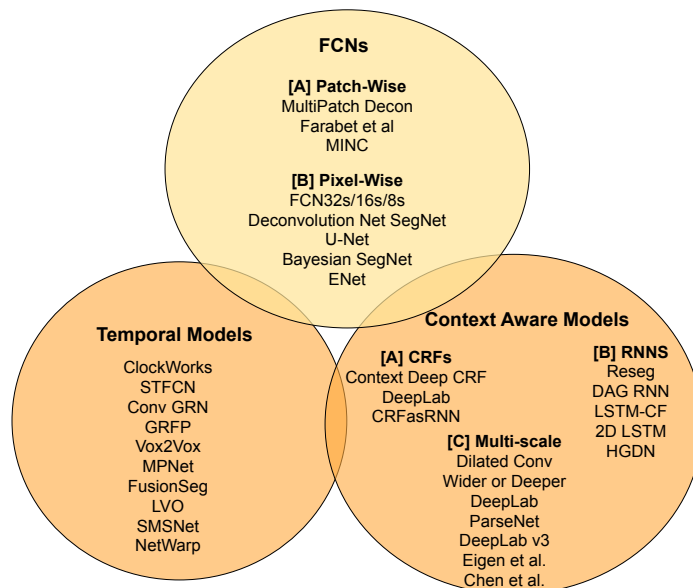


Figure 2.1: Taxonomy of semantic segmentation approaches categorized into (1) Basic fully convolutional networks. (2) Models that incorporate contextual information. (3) Models that incorporate temporal information for video segmentation.

### 2.1.1 Classical Methods

Image segmentation was mainly defined as a way to partition the image into semantically meaningful regions and predicting a class label for each region. A few years ago, it was seen as a challenging problem to achieve reasonable accuracy in semantic segmentation. One perspective of classical image segmentation approaches, is focused on optimization based methods that formulate the segmentation problem as a minimization of a certain cost functional. These methods can be categorized into using spatially discrete [9] or spatially continuous representations [27][26]. In methods that use spatially discrete representations usually the pixels within an image are represented as graph nodes, and different optimization techniques can be used to minimize the cost function and find the minimum cut such as [9]. On the other hand, methods that use spatially continuous representations [26][26] use variational methods to minimize the cost function and evolve contour in the negative gradient

direction to reach the optimal segmentation such as snakes [68].

Popular among graph based methods are conditional random fields (CRFs) [158][80][78]. In CRFS the energy function combines unary and pairwise potentials. The unary potentials give a probability of whether the pixel belongs to a certain class. While pairwise potentials which are also referred to as smoothness term ensures label consistency among connected pixels. A pixel-level or region-level or a hybrid approach can be used to model the pairwise relations. Krahenbuhl et. al. [78] instead proposed a fully connected dense CRF, with an efficient inference algorithm to model long-range connections. In their unary potentials, they relied on boosting [139] classifiers with features combining color, histogram of oriented gradients and pixel location features.

Other approaches for segmentation outside the optimization based methods are Boosting [139] and Random Forests [138][10]. Boosting can be used to classify pixels and perform the final semantic segmentation task. It is based on combining multiple weak classifiers that uses color, shape or texture as in [158][139]. Other methods that relied on random forests, such as Shotton et. al. [138], where each tree was trained on random subset of the training data. These methods implicitly cluster the pixels while explicitly classifying the patch category. Brostow et. al. [10] proposed a randomized decision forest, however instead of using appearance based features, motion and structure features were used. These features include surface orientation, height above camera, and track density where faster moving objects have sparser tracks than static objects. They rely on hand crafted features and perform pixel-wise classification independently without utilizing the structure in the data if used solely. Generally, the performance of classical methods was always bounded by the performance of the hand crafted features used. Even with conditional random fields the unary potentials were constructed based on classifiers that rely on hand-crafted features such as boosting [158]. But that was overcome with deep learning as will be discussed in the following sections. Nonetheless, conditional random fields can still benefit deep learning even if used as a post processing step to ensure smoothness of the semantic segmentation predictions as in [18].



### 2.1.2 Fully Convolutional Networks (FCN)

The initial direction in semantic segmentation using convolutional neural networks was towards patch-wise training to yield the final segmentation. However, in deep semantic segmentation the dominant direction is to learn pixel-wise classification in an end-to-end manner [2], [92], [107]. Long et al. [92] started by proposing fully convolutional networks (FCN). The network learned heatmaps that were upsampled with-in the network using transposed convolution to get dense predictions. Unlike patch-wise training this method used the full image to infer dense predictions which is more computationally efficient as discussed in [92]. The SkipNet architecture was utilized to refine the segmentation using higher resolution feature maps as shown in Figure 2.2. Badrinarayanan et al. [2] proposed SegNet which is an encoder-decoder architecture. The decoder network upsampled the feature maps by keeping the max-pooling indices from the corresponding encoder layer. Kendall et al. [69] followed that work by proposing Bayesian SegNet, which incorporates uncertainties in the predictions using Monte-Carlo dropout during inference. Ronneberger et al. [130] proposed a u-shaped architecture network where feature maps from different encoding layers are concatenated with the upsampled feature maps from the corresponding decoding layers. Figure 2.2 shows the comparison between Skip Architecture and U-Net. Both improve the predicted segmentation through recovering the loss in resolution from the encoder stage. Paszke et al. [113] proposed the use of bottleneck modules for a computationally efficient solution that is called ENet.

### 2.1.3 Context Aware Models

Refinements on fully convolutional networks were introduced to improve segmentation accuracy by incorporating context. In this section we consider only the spatial context, we do not include any temporal information in this category. The methods to enforce models to become context aware are mainly categorized into multi-scale support, utilizing conditional random fields, or recurrent neural networks. Long et al. [92] proposed the skip architecture to

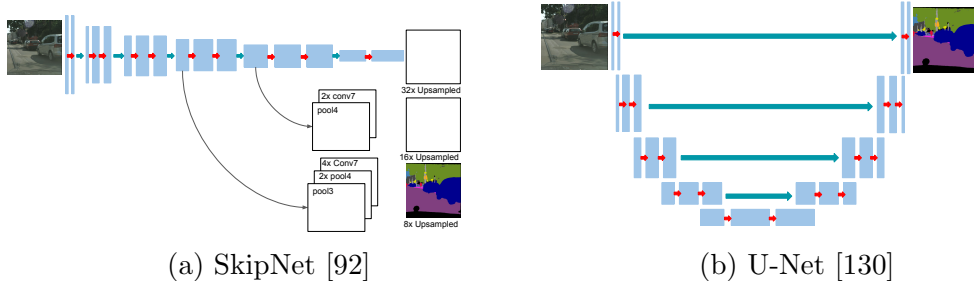


Figure 2.2: Two decoding methods for fully convolutional networks. The decoding method describes the approach for upsampling and computing the output pixel-wise labels. Figures adapted from [92], [130].

merge heat-maps from different resolutions. Since these architectures include pooling layers to increase the receptive field, this leads to loss in resolution from downsampling the image. Yu et al. [199] introduced dilated, or atrous convolutions, which expanded the receptive field without losing resolution based on the dilation factor. This provided a better solution for handling multiple scales. Wu et al. [191] proposed a shallower but wider network using residual connections that included dilated convolution and outperformed deeper models. Chen et al. [18] proposed DeepLab that used atrous spatial pyramid pooling (ASPP) for multi-scale support. The idea was built on utilizing the dilated convolutions. Figure 2.3 shows the proposed ASPP and the dilated convolution. Chen et al. [20] refined further the DeepLab method by incorporating global context features.

One of the commonly used models to incorporate context is conditional random fields (CRF). Chen et al. [18] utilized fully connected conditional random fields as a post processing step. Gaussian kernels based on the spatial and color features were used as pairwise potentials, while the unary potentials were set to the probabilities from convolutional networks. Zheng et al. [204] formulated the mean field CRF inference algorithm as a recurrent network. This method enabled the end-to-end training of the model. In contrast to the previous work that used conditional random fields as a post processing refinement step, this work went further in integrating CNNs and CRFs.

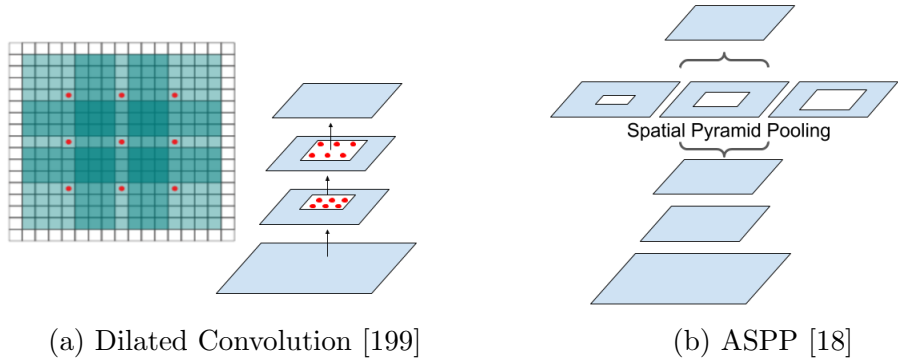


Figure 2.3: Dilated (Atrous) Convolution with a 3x3 kernel with holes and Atrous Spatial Pyramid Pooling. Figures adapted from [18], [199].

### 2.1.4 Video Semantic/Panoptic Segmentation

Recently some approaches emerged for video semantic segmentation that utilized temporal information [42] [106]. Tran et al. [167] proposed a 3D convolutional network trained end-to-end for video semantic segmentation, where 4D kernels are learned on an input of sequence of images. An issue with 3D convolution is its small finite duration on the temporal axis that can not capture long temporal dependencies. Recurrent neural networks can alleviate this. Fayyaz et al. [42] incorporated spatio temporal features by using a layer grid of Long Short term memory models (LSTMs). However, conventional LSTMs that were mentioned earlier do not utilize the spatial coherence and have many parameters to learn. The convolutional gated recurrent unit enabled the network to learn both spatial and temporal information with fewer parameters. Nilsson et al. [106] combined the power of both convolutional gated architectures and spatial transformers for leveraging video semantic segmentation. Spatial transformers were used to warp the previous frame segmentation along the optical flow fields, which is further fused with the current frame estimates using gated recurrent units. An action recognition comparative study [16], showed that two-stream 3D convolution architectures that utilized optical flow information outperform Conv-LSTM models. That motivated more research in the direction of incorporating optical flow for video object/semantic segmentation.

Gadde et al. [45] proposed a method for applying feature warping through

an intermediate module termed as NetWarp in order to incorporate temporal information from videos. The release of the video segmentation benchmark DAVIS [115] provided the means to compare and compete with the state of the art on this challenging problem which is discussed further in Section 2.3. Previous literature focused solely on either temporal consistency of semantic segmentation or tracking segmented instances as in DAVIS’17 [119] through a video sequence. After the introduction of the combined panoptic segmentation task that performs both semantic and instance segmentation, new work in video panoptic segmentation emerged. Kim et. al. [73] proposed a method to perform video panoptic segmentation which uses a spatio-temporal attention module to ensure temporal consistency of the output semantic segmentation. A tracking head that ensures the temporal consistency of instances segmented is also used. The tracking head is composed of multiple fully connected layers that predicts an association vector between the current detected objects and the ones tracked in a memory queue. We continue as well to extend the panoptic segmentation task but rather with class agnostic segmentation through video sequences, where class agnostic segmentation will be discussed in Section 2.4.

## 2.2 Few-Shot Object Segmentation

Few-shot learning (FSL) is the ability to generalize from few examples. It acts in a way as a test-bed for human-like learning [181]. Children are able to grasp concepts and generalize to novel ones from fewer examples [97][12] than what current deep learning methods need. It has the potential to learn from rare cases and to reduce the cost of data collection. A detailed survey [181] introduced an exhaustive discussion of the few-shot learning literature. However, since we focus on few-shot object segmentation we will briefly introduce the FSL taxonomy and the major works in few-shot learning relevant to ours. Then a detailed discussion of few-shot object segmentation will be introduced.

### 2.2.1 Few-shot Learning Setup and Taxonomy

The few-shot learning setup involves classification, detection, or segmentation of a query image  $Q$  based on few labelled training examples called a support set  $S$ . The basic setup on a classification task evaluates  $K$ -shot  $N$ -way classification, where  $K$  indicates the number of training examples in the support set and  $N$  is the number of classes to classify among. Various datasets were proposed to evaluate the few-shot classification task such as Omniglot [81], miniImagenet [173], tieredImageNet [128] and the most recent meta-dataset for few-shot learning [168]. A taxonomy of the few-shot classification following the discussed methods in [22] is shown in Table 2.1. It mainly uses three categories for few-shot learning methods which are: (1) Initialization based methods. (2) Metric learning methods. (3) Hallucination based methods. **Initialization based** methods or *learning to fine-tune* tries to either learn a good initialization to the network parameters or tries to learn an optimizer. **Metric learning** methods *learn to compare* between the support set and query images with methods using meta-learning detailed in Section 2.2.2 and others based on cosine classifiers that measure the cosine similarity between feature representation and classification weights. It is worth noting that some of the cosine classifiers methods use meta-learning schemes such as Gidaris et. al. [49]. **Hallucination based methods** *learn to augment* the few labelled data to improve the method generalization. It is worth noting that most of the methods in different categories are meta-learning methods as well. Thus, meta-learning can be looked upon as a topic that overlaps with few-shot learning methods. It was shown in [22] that using cosine classifiers similar to the work by [120][49] performs on par to meta-learning methods and even outperforms them when training from one domain to another.

### 2.2.2 Problems Related to FSL

There are some terms that relate to few-shot learning that we want to clarify to demystify any confusions namely: (1) Meta-learning. (2) Zero-shot learning. (3) Imbalanced training. (4) Generalized few-shot/zero-shot learning. (5)

Table 2.1: Few-Shot Learning Taxonomy.

Category	Sub-Category	Methods
Initialization Based	Good Model Initialization	Finn et. al. [43][44] Nichol et. al. [105] Rusu et. al. [132]
	Optimizer	Ravi et. al. [123] Munkhdalai et. al. [102]
Metric Learning	Meta-Learning	Koch et. al. [76] Vinyals et. al. [173] Snell et. al. [153] Bertinetto et. al. [7] Sung et. al. [160] Gidaris et. al. [49]
	Cosine Classifiers	Qi et. al. [120] Gidaris et. al. [49] Chen et. al. [22]
Hallucination Based		Wang et. al. [180] Antoniou et. al. [1] Hariharan et. al. [54]

Open-set recognition. Meta-learning which is learning to learn [43][44], learns from a sampled set of tasks in the training phase to generalize to the novel tasks in the inference phase. It is more of a scheme to train the model to generalize to novel tasks which are structurally similar to the tasks sampled during training. When applied to the few-shot learning problem a set of sampled tasks are used in the training by sampling both support set and query images labelled with  $L_{train}$  this is called meta-training phase. During the meta-testing phase the model is tested on a sampled support set and query images with labels from  $L_{test}$ . This meta-training phase simulates the few-shot setting and enables the model to better generalize to novel classes. Meta-learning has also been used in neural architecture search [89] not necessarily the few-shot learning problem solely. On the other hand not all few-shot learning methods follow a meta-learning scheme in their training. Some learn the parameters directly [120] or perform it on two training stages one is regular network training followed by a meta-training stage [49][159].

Other related terms to few-shot learning is zero-shot learning which learns to classify unseen classes using their textual descriptions. Unlike few-shot

Table 2.2: Summary of Different Related Tasks to Few-shot Learning and their differences. Replica of Table from [48].

Task Setting	Training	Testing	Goal
Traditional	KKC	KKC	Classifying KKC
Reject Option	KKC	KKC	Classifying KKC and rejecting low conf. samples
Few-shot	KKC & few UKC	UKC	Identifying UKC
Generalized Few-shot	KKC & few UKC	KKC & UKC	Identifying KKC & UKC
Zero-shot	KKC & side info.	UKC	Identifying UKC
Generalized Zero-shot	KKC & side info.	KKC & UKC	Identifying KKC & UKC
Open-Set	KKC	KKC & UUC	Identifying KKC & rejecting UUC

learning in which the support set still has few labelled samples for the novel classes, zero-shot learning has text description instead. Imbalanced learning [28] is also a close problem to the few-shot learning problem. It addresses the problem of learning classes with abundant data along other classes with fewer labelled data. However, few-shot learning provides an extreme imbalance scenario where only one, five or maximum twenty samples are provided for the novel classes. Also in imbalanced training the luxury of training with both data points exists. In few-shot learning most methods tend to learn on the base classes with large-scale data, then is expected to generalize from few-shot support set to novel classes without retraining on both. Generalized few-shot and zero-shot learning are both extensions of their corresponding tasks where the main difference is that during inference evaluation on both base classes and novel classes is performed. Open-set recognition is another relevant task that is concerned generally with identifying unknown classes without any extra/side information provided for these unknown classes.

Table 2.2 from [48] summarizes these different related tasks based on their goal and what is provided during training and testing. Based on the above problems all classes can be classified into three different categories:

- Known Known Classes (KKC), also called base classes.
- Unknown Known Classes (UKC), no available samples during training except limited or side information also called novel classes.
- Unknown Unknown Classes (UUC), classes without any information during training or testing.

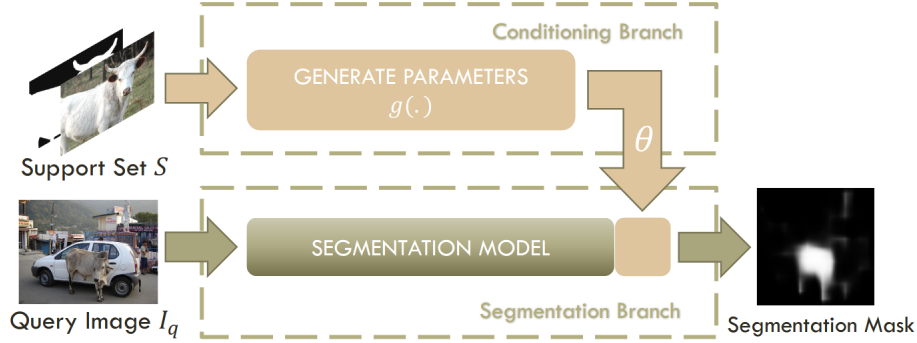


Figure 2.4: OSLSM method for few-shot object segmentation. Figure from [136].

### 2.2.3 Few-shot Object Segmentation Setup and Metrics

Although few-shot learning was initially focused on the object classification task, few-shot object segmentation emerged recently due to the various robotics applications that requires scene understanding such as autonomous driving [25][131] and robot manipulation[70][32]. Deep semantic segmentation has been rigorously studied in the literature [92][199][19]. However, the focus on semantic segmentation using large-scale datasets hinders its use in robotics applications where it can encounter novel objects never seen before [117]. The first attempt to perform few-shot object segmentation by Shaban et. al. [136] introduced a widely used few-shot object segmentation benchmark called Pascal-5<sup>i</sup> [136]. The 20 classes from PASCAL-VOC [40] are split into 4 folds with 5 classes each. A similar setup was performed on MS-COCO [177][201]. The setup on Pascal-5<sup>i</sup> is a 1-way k-shot segmentation, where the goal is to segment the class of interest against the background. There are two evaluation metrics used which are mean intersection over union [136] and binary intersection over union [121]. The later averages the mean intersection over union for foreground and background. While the former computes the mean intersection over union over all classes. Although it is a 1-way setup as mentioned earlier, we compute a per-class foreground intersection over union then average over all classes.



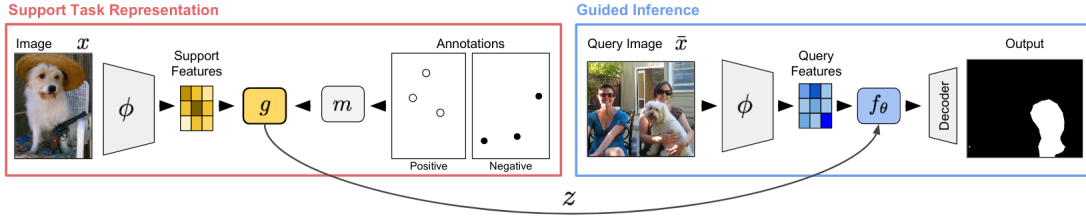


Figure 2.5: CoFCN method for few-shot object segmentation. Figure from [121].

## 2.2.4 Few-shot Object Segmentation Literature

Shaban et. al. [136] used a two-branch method as shown in Figure 2.4, where the first branch is responsible for foreground/background segmentation. The second branch takes the support set image-label pairs and performs weight hashing to predict the weights for the novel class segmentation. Three different baselines were introduced in it, which are: (1) Base classifiers: which trains FCN32s [92] model on 16 classes outside the fold and extracts the features to be used with either k-nearest neighbour or logistic regression. (2) Siamese networks that measures pixel L1 similarity and uses a binary cross entropy loss on the output similarity measured during training. In the inference phase each pixel in the query image is classified based on the nearest neighbour to the trained distance metric. (3) Fine-tuning the model as a simple baseline. Another work by Rakelly et. al. proposed another 2-branch method where the second branch acts as a conditioning branch on the masked representation of the support set instead of predicting parameters [121] as shown in Figure 2.5. Their early fusion strategy requires learning two separate sets of parameters; however, their late fusion strategy can share weights. Their proposed model allows for training on sparse annotations that enables interactive segmentation with simple clicks on some pixels. Dong et. al. inspired by prototypical networks, designed a method to learn prototypes for the few-shot object segmentation problem [33]. The model consists of a 2-branch architecture, where the second branch is responsible for learning prototypes.

Concurrent to our work, four methods were recently proposed [203][201][177][200]. Zhang et. al proposed a single branch network that uses guidance features

based on masked average pooling layer [203] which makes it more efficient than two-branch methods. Zhang et. al. proposed a class agnostic network to perform few-shot object segmentation using a dense comparison module and an iterative optimization module [201]. Figure 2.6 shows detailed architecture for the proposed CANet method, where the dense comparison module uses the output from masked average pooling which can be looked upon as a prototype as well. However, the output goes through atrous spatial pyramid pooling and an iterative optimization module that takes the input predictions from previous time step and learns the new improved prediction while using residual connection. As shown in Equation 2.1

$$M_t = x + F(x_t, y_{t-1}), \quad (2.1)$$

where  $M_t$  is the current prediction step and  $y_{t-1}$  is either set to the initially predicted logits or the previous step prediction. The output feature map  $x$  is the output from the comparison module, and  $F$  is simple concatenation of  $x, y_{t-1}$  followed by  $2 \times 3 \times 3$  convolutions. The atrous spatial pyramid pooling that allows to incorporate different contextual information along with the iterative optimization module improves the final segmentation accuracy. Wang et. al. [177] proposed a prototype alignment method that performs support-to-query and query-to-support segmentation. In a way it ensures the output prototypes are aligned for both support and query and can be used to improve any method that builds on prototypes. Finally, Zhang et. al. in another work [200] proposed to use a pyramid of graph attention units. Each unit relates different nodes from support to the query to propagate labels from the support set. A pyramidal approach to work on multi-resolution levels is proposed to improve the segmentation accuracy.

It can be seen that most few-shot object segmentation work depends on the idea of using prototypes in one way or another. All of this previous literature except for [200] assumes a single vector prototype to be representative enough for the support set through masked average pooling. Although, Zhang et. al. [200] proposed a graph based approach to propagate the pixel labels

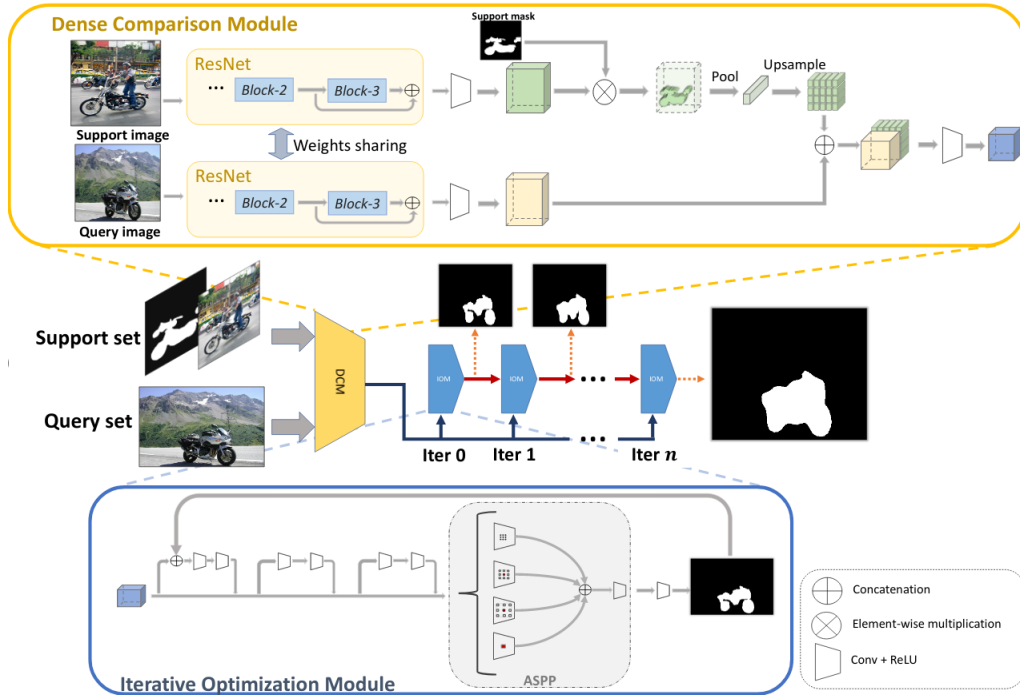


Figure 2.6: CANet method for few-shot object segmentation. Figure from [201].

from support set to query. We were the first to propose without pixel-level labels but rather depending on image-level labels to leverage the interaction between support and query images through an attention based mechanism. We were also one of the first to propose a single branched few-shot object segmentation method and improve the segmentation accuracy through the idea of multi-resolution imprinting, unlike other methods that relied on two-branched approaches.

## 2.3 Video Object Segmentation

Video object segmentation has two main categories which are unsupervised and semi-supervised methods. Unsupervised methods deal with the problem of segmenting the primary object in the video sequence in terms of its appearance and motion saliency. While semi-supervised approaches are expected to track the masks provided in the first frame through the video sequence. Both problems share some challenges such as object occlusions, cluttered back-

Table 2.3: Video Object Segmentation(VOS) Taxonomy.

Category	Sub-Category	Methods
Unsupervised VOS	Flow Based	Tokmakov et. al. [165][166] Jain et. al. [66]
	Attention Based	Lu et. al. [94] Wang et. al. [178] Yang et. al. [196]
	Others	Koh et. al.[77] Song et. al. [154] Seguin et. al. [135]
Semi-supervised VOS	Fine-tuning	Caelles et. al. [13] voigtlaender et. al.[176] Hu et. al. [64] Luiten et. al.[95]
	No Fine-tuning	Voigtlaender et. al.[174] Wang et. al.[182] Yang et. al. [195]

ground, drastic appearance changes of the object. The unsupervised method has some relations to motion segmentation since motion plays an important role in determining the primary object. Table 2.3 shows the general taxonomy of video object segmentation methods. The most widely used datasets in the literature for video object segmentation are DAVIS [116][119], SegTrack [84], FBMS [108], and Youtube-VOS [194]. In the next sections we will cover the details of these three related tasks. The main metrics [116] used for evaluating video object segmentation are mean intersection over union which is referred to as region similarity. F-measure is also used which is referred to as contour accuracy. Finally, temporal stability is used to ensure consistency of segmentation throughout the video sequence. Each mask is transformed into polygons and for each point on the polygon shape context descriptors are computed. Then a dynamic time warping technique is used to match points across frames.

### 2.3.1 Unsupervised Video Object Segmentation (UVOS)

Unsupervised video object segmentation methods [77][165][66][166] do not rely on any initialization mask and rather depend on segmenting the primary ob-

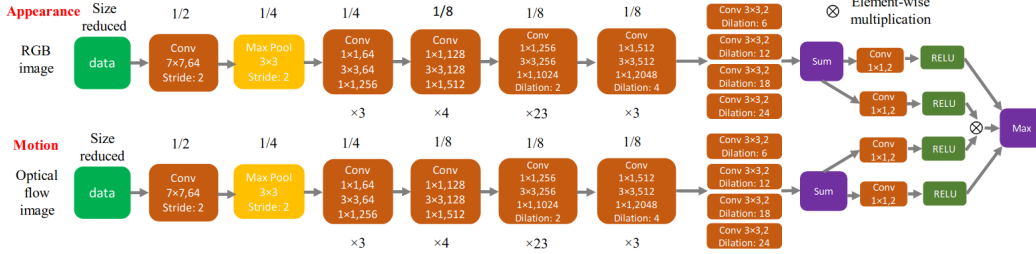


Figure 2.7: FusionSeg method Figure from [66].

ject in the scene. The primary object is determined based on both appearance and motion. Classical approaches rely on building a spatio temporal graph [135]. Koh et. al.[77] presented a method based on extracting candidate proposals for the primary object. Then followed this with augmenting or reducing the regions proposed for a finer segmentation. Since this method is based on handcrafted features from color and motion edges to propose the regions, it is prone to error unlike end-to-end trained deep networks. Prominent deep learning based methods rely either on optical flow [165][166][66] or attention [196][178][94]. Flow based methods, such as the work from Jain et. al. [66], combined motion from optical flow and appearance in a two-stream fully convolutional network. Figure 2.7 shows their architecture for video object segmentation, a method to generate large weakly annotated data for training the motion stream was proposed as well. Tokmakov et. al. [166] improved the idea further through learning a visual memory module using bidirectional convolutional gated recurrent units thus enabling the segmentation of the primary object throughout a video sequence even if it becomes static in certain frames through the memory module. These previous works represent good direction in incorporating motion cues.

On the other hand attention based methods, such as the work from Lu et. al. [94] proposed a method to use co-attention Siamese network to learn correlation between randomly sampled frames to discover a primary object that persists throughout all the sequence. Figure 2.8 shows the detailed architecture for their method. In addition to that, the formulation as a comparison between randomly sampled frames allows for increasing the training data samples. Wang et. al. [178] proposed using attention as a guide to unsupervised

video object segmentation. They started with extending DAVIS and other VOS datasets with eye fixation annotations which was recorded using a SMI RED250 eye tracker. Then a model built on a dynamic UVOS-aware module that predicts visual attention using a convolutional LSTM is used to guide the segmentation method. Yang et. al. [196] combines self attention and attention conditioned on the anchor embeddings, where the anchor is the first frame of the video sequence, to perform UVOS. In our opinion, a combination of using motion and depth cues along with attention mechanisms to leverage the interaction among frames is the best direction as some methods [73] explored the use of spatio-temporal attention along with warping features using optical flow. However, the previous method does not aim at segmenting primary objects in a sequence but rather a slightly different task as explained in Section 2.1.4.

Other methods not relying on explicit motion representation as optical flow but rather relying on recurrent models to learn implicit motion representation such as [154]. Song et. al. [154] proposed a pyramid dilated convolutional LSTM module that extracts spatial features on multiple resolution levels within a convolutional LSTM module. The main purpose is to capture global context through the module responsible for implicitly representing motion, which learns spatio-temporal saliency object segmentation. Most of the unsupervised VOS methods perform binary segmentation without considering segmenting different instances, except for Seguin et. al. [135]. In our proposed methods we rather invest in considering both video semantic/panoptic segmentation along with class agnostic segmentation that perform generic object segmentation regardless of semantics. It is not only considering visually salient objects nor does it only consider objects within a closed set of known classes as detailed in Section 2.4.

### **2.3.2 Semi-supervised Video Object Segmentation (SVOS)**

Semi-supervised video object segmentation relies on an initialization mask for the first frame. A prominent direction in SVOS relies on fine-tuning or adapting the model online with each video sequence to predict the segmenta-

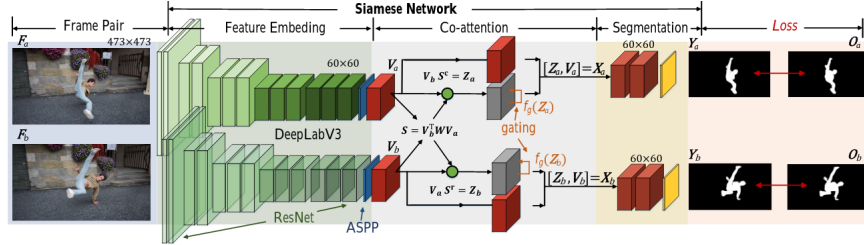


Figure 2.8: Coattention for Video Object Segmentation method Figure from [94].

tion [13][176][95]. Caelles et. al. [13] presented a one shot video object segmentation work that fine-tunes the network based on the initial mask of the sequence. Voigtlaender et. al. [176] followed the same idea with on-line adaptation. The method adapts the network to aid it in learning the appearance changes of the object. The purpose of on-line training is to get the network to capture the appearance of the object being segmented in the current sequence. Luiten et. al. [95] proposed to generate different object proposals using a class agnostic mask R-CNN and is further refined to predict masks with a network fine-tuned separately for each sequence.

Methods generally relying on fine-tuning on the first frame or performing online adaptation of the model suffer from being computationally inefficient. Other methods [174][195][64][182] proposed different ways to address semi-supervised video object segmentation without online adaptation. Voigtlaender et. al. [174] proposed a method to perform global and local matching using the learned pixel embeddings. Global matching matches the current frame embeddings to the first frame while local one matches it to the previous frame, this matching of the learned pixel embeddings alleviates the need to perform fine-tuning. Yang et. al. [195] proposed MaskTrack which is based on Mask R-CNN [56] with an added tracking head that used a multi-class formulation to the tracking problem. Wang et. al. [182] proposed a ranking attention module that selects important features, it was used in a method that combines the power from both matching based and propagation based methods.

### 2.3.3 Motion Segmentation

Motion segmentation is a fundamental problem in computer vision and robotics, with a long history [155][198]. Multiple approaches whether geometry or learning based were explored in the literature. Scott proposed a geometry based work that models the background motion in terms of homographies [183]. It is based on the limiting assumptions that either the background is mostly planar or the camera motion is mainly rotation. Both assumptions can lead to failures in the case of general camera motion in autonomous driving scenes. Some techniques rely on object tracking to generate trajectories that can be used further for motion segmentation such as [11], but they tend to be computationally inefficient. Reddy et al. proposed the use of fully connected conditional random fields for the joint prediction of semantics and motion labels [126]. The approach is computationally inefficient and runs at 240 seconds as reported in [172] on images at the resolution of 348x768.

The motion segmentation problem relates to the unsupervised video object segmentation problem since the primary object is the most salient object in terms of appearance and motion. Tokmakov et al. used a one-stream fully convolutional network with optical flow input to estimate the motion type [165]. Neglecting appearance information and relying only on motion solely can lead to degraded accuracy compared to the combined information in autonomous driving specifically. Drayer et al. described a video object segmentation work that used tracked detections from R-CNN denoted as tubes [37]. The main issue with this approach is its running time of 8 seconds per frame. Concurrent to our first moving object detection network for autonomous driving, Vertens et al. proposed a network that yields pixel-wise semantic motion labels [172] that is based on the Flownet-2 architecture [65]. They performed ego-flow suppression to compute the flow for moving objects solely and use that as input to their moving object segmentation network. In contrast to the previous methods we focus on real-time performance, multi-task learning of both class agnostic and semantic heads, along with learning instance-wise motion masks. Concurrent to our work a recent work by Lee et. al. [83] explored motion



segmentation on the level of instances through learning an ego-poseNet and an obj-poseNet to separate both motion trained in a self-supervised manner through instance-wise reconstruction and geometric consistency losses. However, their method does not explore multi-task learning of semantic and motion instance segmentation and does not focus on real-time performance.

## 2.4 Intersection between FSS and VOS

Although the fundamental question in both few-shot object segmentation and video object segmentation is different, it turned out to be extremely beneficial to learn from both topics interchangeably. The core of the problem in few-shot object segmentation that needs to be addressed is the generalization ability to novel classes from few labels after learning from large-scale data for the base classes. On the other hand, the core problem in unsupervised video object segmentation is identifying the object of interest, whether primary (unsupervised) or based on an initialization mask (semi-supervised), throughout the video sequence and adapting to the appearance, illumination changes in the following frames. However, if we solely focus on the research questions without considering the context of what raised these questions it can lead us to non-optimal local minimas in our thinking of the problem. If the context to study the few-shot object segmentation problem is to test human-like intelligence, then it is necessary to consider other aspects such as incorporating motion and geometry. It also entails studying temporal consistency of the representation which can aid the few-shot object segmentation problem. This can lead to an interesting relation between both topics.

The semi/un-supervised video object segmentation are also called few/zero-shot VOS. In both video and few-shot object segmentation the problem of learning correspondences between support set (initialization mask or the segmented primary object) and the query image (consecutive frames of the video sequence) is crucial. Leveraging the interaction between the support and query brings an interesting challenge in FSS that mainly inspires from VOS counterpart [94]. These overlapping benefits motivated our direction to study both

Table 2.4: Summary of FSS and VOS overlap. FSS: Fewshot segmentation. VOS: Video object segmentation. VCAS: video class agnostic segmentation.

	FSS	VOS
<b>Shared Challenges</b>	How to generalize to novel classes with few training data?	How to generalize to different appearance changes from illumination, occlusion, deformations from few labelled initialization whether manually labelled (Semi-VOS) or automatically segmented (Un-VOS)?
	How to leverage interaction between query and support sets?	How to leverage interaction between reference frame and consecutive frames in the sequence?
<b>Shared Assumptions &amp; Solutions</b>	Attention mechanisms help to learn better interaction between support and query - Chapter 4.	Attention mechanisms help learn better interaction between sequence frames [94][196].
	Pixels that move together belong to same object can aid the learning from few labelled data - Chapter 4.	Same assumption can aid with temporal consistency of objects segmentation through the video sequence.
	Learning to compare (metric learning) improves the representation learned for few-shot task - Chapter 3.	Learning to compare improves the representation to discriminate among known and unknown objects - Chapter 5.

problems simultaneously.

One of the tasks that was overlapping between both as shown in Table 2.4 benefits from FSS and VOS is video class agnostic segmentation. The goal of video class agnostic segmentation in autonomous driving is to segment objects of unknown classes towards a safety critical approach. The table summarizes the shared challenges and assumption in both areas and how video class ag-

nostic segmentation is a shared problem and benefits from both. We focus on combining appearance, motion and geometry with a different learning objective aimed at segmenting generally moving objects, or objects outside the closed known classes using our proposed contrastive losses. We call our task video class agnostic segmentation as it is in-dependant of the semantics of these unknown objects. The video class agnostic segmentation is crucial for robotics tasks such as autonomous driving or robot manipulation in which rare cases / object categories are expected to occur and should be handled in the system.

## 2.5 Background

### 2.5.1 Transposed Convolution Arithmetics

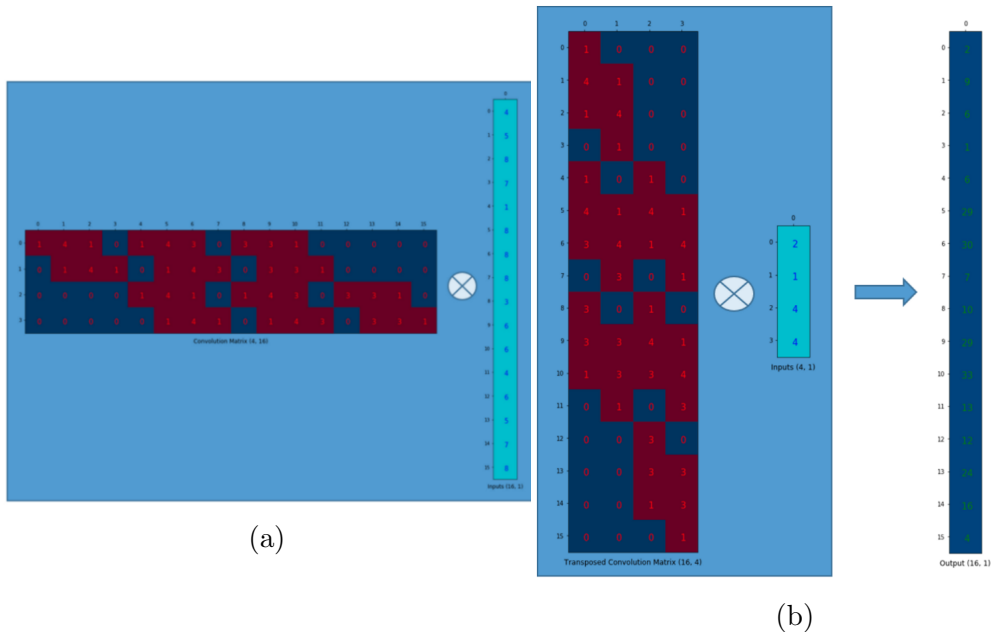


Figure 2.9: (a) Convolution , (b) Transposed Convolution Operation as matrix multiplication, Figure from [103].

End-to-end models for semantic segmentation incorporate in-the-network upsampling that is performed using transposed convolution. In order to understand the transposed convolution, it is illustrative to formulate the regular convolution as matrix multiplication. Note that convolution is implemented as

matrix multiplication in modern deep learning frameworks for efficiency reasons as well. As an example, convolution with  $3 \times 3$  kernel  $W$ , is represented as a sparse matrix  $c$  with the non zero elements as the kernel coefficients  $w_{i,j}$ . The sparse matrix is multiplied with the flattened input. Thus a  $4 \times 4$  input matrix is flattened to a 1D vector of 16 elements and multiplied by  $c$ . The convolution output is a 1D vector of 4 elements that is reshaped to the output feature map of  $2 \times 2$  shape as shown in Figure 2.9 (a).

$$c = \begin{pmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{pmatrix} \quad (2.2)$$

In-the-network upsampling is performed in the reverse direction where an input vector of shape  $4 \times 1$  is upsampled to  $16 \times 1$ . This can be done by multiplying with  $c^T$ , hence the name transposed convolution as shown in Figure 2.9 (b). Since the backward pass through regular convolution requires multiplying with  $c^T$ , the transposed convolution has it reversed. Where the forward pass multiplies by  $c^T$  and the backward pass multiplies by  $c$ . Further details are described in [38].

## 2.5.2 Learning Based approach for Optical Flow

Optical Flow classical approaches such as Lucas Kanade and Horn Schunck has been widely used in different applications, but with the need for accurate and computationally efficient inference of dense optical flow, learning based approaches emerged. FlowNet [34] is considered the first approach to tackle this problem. In FlowNet a two-stream model with a correlation layer that performs patchwise correlations. As shown in Equation 2.3

$$c(\mathbf{x}_1, \mathbf{x}_2) = \sum_{i \in [-k, k] \times [-k, k]} \langle f_1(\mathbf{x}_1 + i), f_2(\mathbf{x}_2 + i) \rangle, \quad (2.3)$$

where  $k$  is patch size,  $f_1, f_2$  are the two feature maps and  $x_1, x_2$  are two centers at both feature maps. The operation is similar to a convolution but instead of using a learnable filter it uses the other feature map patch. Then, the model predicts the x-y flow outputs and is trained as a regression problem. Learning

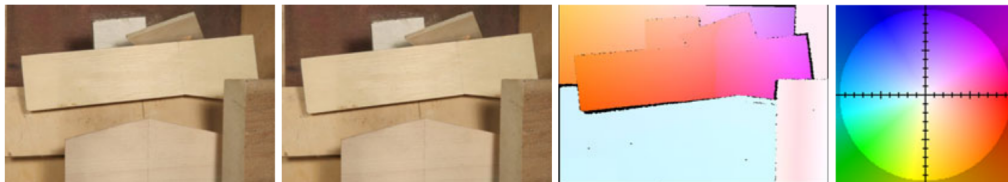


Figure 2.10: Flow HSV Encoding, Figure from [5].

based approaches require large-scale training data, however it is not trivial to have optical flow ground-truth. For this purpose multiple synthetic datasets such as FlyingChairs [34] and FlyingThings3D [98] were proposed to train these models. In Flownet 2.0 [65] a learning schedule with multiple datasets is proposed to improve the optical flow accuracy. Where they initially train the model with FlyingChairs dataset, then FlyingThings3D dataset. FlyingChairs dataset consists of simple 2D motion that enables the model to learn the concept of matching. While FlyingThings3D dataset has 3D motion to learn complex motion patterns. Another important modification in Flownet 2.0 is the introduction of iterative refinement in deep networks through stacking flownet modules and warping the image with the output flow from each stage. Throughout our work we mainly use Flownet 2.0 [65] and then transform the predicted flow to RGB image using middlebury color wheel [5] as shown in Figure 2.10.

### 2.5.3 Self Supervised Learning of Depth and Pose

In this section we introduce self supervised learning of depth and pose as it will be used in our method to perform ego-flow suppression. It was initially proposed by Zhou et. al. [205] to learn depth and pose in a self supervised manner from a monocular camera by posing the problem as a novel view synthesis problem. We use the recent work from Godard et. al. [50] that learns from both monocular and stereo cameras and handles occlusion through back-propagating the minimum reprojection error. The model consists of separate depth and pose networks where the pose network estimates 6 DoF transformation and the depth network estimates per-pixel inverse depth (disparity) as shown in Figure 2.11.

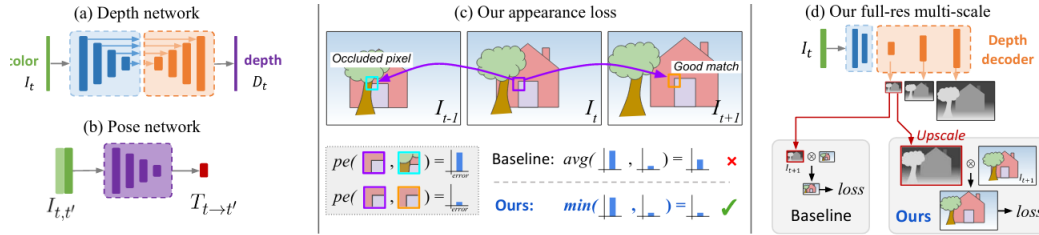


Figure 2.11: Self Supervised Depth and Pose, Figure from [50]. (a) Depth Network. (b) Pose Network. (c) Photo-metric re-projection error, when handling occlusions the average loss will force occluded pixels to match, whereas the minimum loss only matches it to the visible.(d) Multi-scale.

A view synthesis loss shown in Equation 2.4 is used to minimize the photometric re-projection error. Where  $I_{t'}$  is the reference view and  $I_t$  is the target view,  $\alpha$  is the photometric error,  $\langle$  is the sampling operator,  $\phi$  is the projection function that outputs 2D coordinates for the projected depth  $D_t$  in  $t'$ . The estimated pose between two frames is  $T_{t \rightarrow t'}$  and  $K$  is the camera intrinsics matrix.

$$L = \min_{t'} \alpha(I_t, I_{t' \rightarrow t}) \quad (2.4)$$

$$I_{t' \rightarrow t} = I_{t'} \langle \phi(D_t, T_{t \rightarrow t'}, K) \rangle$$

Thus the model is able to jointly learn the pose and depth which we later use to perform ego-flow suppression to remove the optical flow of the non-moving pixels. Their main contribution is the use of minimum photometric error instead of the average over all target views which intuitively leads to handling occlusion problems that should not contribute to the backpropagated loss. Since they use consecutive frames and opposite stereo frames in their mixed training it learns better self supervised models.

# Part I

## Few-shot Object Segmentation

# Chapter 3

## Imprinting Masked Prototypes

### 3.1 Introduction

The first research question we answer in part I from the thesis is “how can deep semantic segmentation rely on limited labelled data for learning novel classes instead of large-scale labelled data?”. Children are able to adapt their knowledge and learn about their surrounding environment with limited samples [97]. One of the main bottlenecks in the current deep learning methods is their dependency on the large-scale training data such as PASCAL-VOC [40] with 20 classes and MS-COCO [87] with 80 classes. However, the number of object categories they cover is still limited despite the significant sizes of the data used. The limited number of annotated objects with pixel-wise labels included in existing datasets restricts the applicability of deep learning in inherently open-set domains such as robotics [30], [117]. This motivated the emergence of few-shot learning methods [76], [124], [133], [153], [173][149]. As formally defined in FSL literature, such as [149], the problem of few-shot learning is how to generate the weights for the novel classes based on few training examples (support set) for these novel classes. These early works were primarily focused on solving few-shot image classification tasks, where a support set consists of a few images and their class labels.

However, semantic segmentation has more relevance to robotics than image classification [25], [32], [70], [131]. Therefore, we focus on few-shot object segmentation. Previous methods in the literature that address the problem of few-shot object segmentation require the training of an additional branch



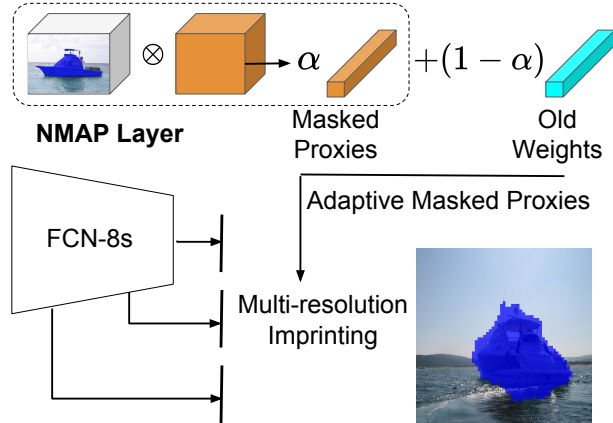


Figure 3.1: Multi-resolution adaptive imprinting in AMP (Adaptive Masked Proxies). Imprinting occurs through masked average pooling of the final activations to be used as the novel class weights. The weights of the background class is further adapted.

to guide the backbone segmentation network, unlike ours that only need a single branch. On top of that, our proposed method outperforms previous SOA methods at the time of the work on Pascal-5i.

### 3.1.1 Metric Learning

Inspired by methods that relate metric learning to softmax classification [120], we propose to construct the weights of the final segmentation layer via multi-resolution imprinting. Our method does not rely on a second parameter estimation branch, as shown in Fig. 3.1. Metric learning methods such as neighbourhood component analysis [51] learn a distance metric using a softmax-like loss function. As shown in Equation (3.1):

$$L_{proxy}(x) = -\log \frac{\exp(-d(x, p(x)))}{\sum_{p(z) \in p(Z)} \exp(-d(x, p(z)))}, \quad (3.1)$$

where  $x$  is a data point,  $p(x)$  is the positive proxy corresponding to the class label of  $x$ , and  $p(Z)$  is the set of negative proxies. The distance used is based on the L2 distance  $d(x_1, x_2) = \|x_1 - x_2\|_2^2$ . In the case of unit vectors minimizing the squared L2 distance becomes equivalent to maximizing the dot product,

$$\min \|x - p(x)\|_2^2 \equiv \max x^T p(x), \quad (3.2)$$

when substituting in Equation (3.1) we get a similar form to softmax classification except that we normalize with all classes logits not negative class logits only. As shown in Equation (3.3):

$$L(x, p_k(x)) = -\log \frac{x^T p_k(x)}{\sum_{c \in C} x^T p_c(x)}, \quad (3.3)$$

where  $p_k(x)$  is the proxy for class  $k$ , while  $p_c(x)$  is the proxy of class  $c$  in the set of all classes  $C$ . Qi et. al. [120] has used this intuition to look upon the final classification weights as proxies. Wu et. al. [188] has also discussed relations between metric learning and softmax classification and looked upon the final classification weights as prototypes. Thus, we use the terms proxies and prototypes interchangeably to indicate a representative signature of a given class that can act as its weight in the classification layer.

In the few-shot segmentation setup, the support set contains pixel-wise class labels for each support image. Therefore, the response of the backbone fully convolutional network (FCN) to a set of images from a given class in the support set can be masked by segmentation labels and then average pooled to create a proxy/prototype for this class. This forms what we call a normalized masked average pooling layer (NMAP in Fig. 3.1). The computed proxies are used to set the  $1 \times 1$  convolutional filters for the new classes, forming the process known as weight imprinting [120]. Multi-resolution weight imprinting is proposed to improve the segmentation accuracy of our method. Our method is a single branch method that does not require extra parameters for learning prototypes or predicting parameters. At the same time our method is the first that can be extended to the generalized few-shot setting, where we can segment among both novel and base classes. Concurrent work to ours, Zhang et. al proposed a single branch network that uses guidance features based on masked average pooling layer [203]. It has also been shown that cosine (cosine similarity) classifiers generally outperform meta-learning methods in [22] which motivated our choice of cosine classifiers in few-shot object segmentation.

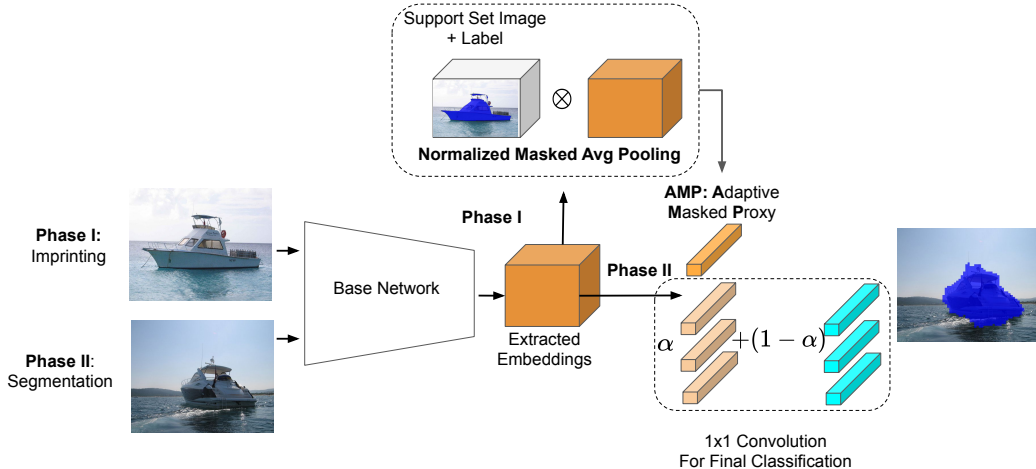


Figure 3.2: Adaptive Masked Proxies using the Normalized Masked Average Pooling Layer. During few-shot inference two phases are computed: (1) Imprinting: which uses the support set image label pair. (2) Segmentation: which predicts the query image segmentation using the assigned weights from the first phase. For simplicity it shows the imprinting on the final layer solely. Nonetheless, our scheme is applied on multiple resolution levels.

## 3.2 Proposed Method

### 3.2.1 Few-shot Problem Setup

We use a setup similar to Shaban et. al. [136]. The training procedure consists of two steps. The initial training phase relies on a large scale dataset  $D_{train}$  including semantic label maps for classes in  $L_{train}$ . During the test phase, a support set is sampled that is labelled with novel classes in  $L_{test}$ , where  $L_{train} \cap L_{test} = \emptyset$ . The support set contains pairs  $S = (I_i, Y_i(l))_{i=1}^k$ , where  $I_i$  is the  $i^{th}$  image in the set and  $Y_i(l)$  is the corresponding binary mask. The binary mask  $Y_i(l)$  is constructed with novel class  $l$  labelled as foreground while the rest of the pixels are considered background. While  $k$  indicates the number of images provided in the support set. A query image is randomly sampled from the test set in a similar fashion to the support set. It is worth noting that during training only images that include at least one pixel belonging to  $L_{train}$  are included in  $D_{train}$  for large-scale training. If some images have pixels labelled as classes belonging to  $L_{test}$  they are ignored and not used in the back-propagation following the procedure from [192]. Our model does not need to

be meta-trained in the few-shot regime by sampling tasks with a support set and a query image, but is rather trained regularly with  $D_{train}$  and classes in  $L_{train}$ .

### 3.2.2 Base Network

The backbone architecture used in our segmentation network is a VGG-16 [151] that is pre-trained on ImageNet [31]. Similar to the FCN8s architecture, [92] skip connections are used to benefit from higher resolution feature maps, and  $1 \times 1$  convolution layers are used to map from the feature space to the label space. However, unlike FCN8s we solely utilize bilinear interpolation layers with fixed weights for the upsampling. The main reason behind that choice, is that it is hard to imprint the weights for the transposed convolution layers based on the support set.

An extension to the above base network uses dilated convolution [199] and called Dilated-FCN8s. The last two pooling layers are replaced by dilated convolution with dilation factors 2 and 4 respectively. Thus, increasing the receptive field without affecting the resolution and improving the segmentation accuracy. Finally, a more compact version of the network with two final convolutional layers removed is denoted as Reduced-DFCN8s.

### 3.2.3 Weight Imprinting

The relation between metric learning and softmax classification has been investigated in [120]. A proxy-NCA loss [101] was reformulated as a softmax cross-entropy loss based on the equivalence between minimizing the Euclidean distance and maximizing dot product of normalized vectors. It motivated the use of the output proxies for each class as the weights of the final fully connected layer for image classification, which is known as weight imprinting. Since  $1 \times 1$  convolutional layers are equivalent to fully connected layers, we propose to utilize proxies to imprint the  $1 \times 1$  convolutional filters of the final segmentation layer. The imprinted convolution weights form a signature for each class. When convolved with the query image, it activates pixels maximally similar to that class signature.

However, it is not trivial to perform weight imprinting in semantic segmentation, unlike in classification. First, in the classification setup the output embedding vector corresponds to a single class and hence can be used directly for imprinting. In contrast to that, segmentation network outputs 3D embeddings, which incorporate features for a multitude of different classes, both novel and previously learned. Second, in the classification scenario the resolution aspect is not present, while in the segmentation scenario multi-resolution support is necessary to ensure the final segmentation accuracy.

We propose the following novel architectural components to address the challenges outlined above. First, in Section 3.2.4 and in Section 3.2.5 we propose the proxy masking and adaptation methods to handle multi-class segmentation. Second, in Section 3.2.6 we propose a multi-resolution weight imprinting scheme to maintain the segmentation accuracy during imprinting. The significant contribution of each method to the overall accuracy is shown experimentally in Section 4.3.3.

### 3.2.4 Normalized Masked Average Pooling

In order to build the proxies and incorporate the pixels that belong mainly to the novel class, masked feature maps with the labels provided in the support set are used. Initially, the feature maps are bilinearly upsampled before performing masking. This is followed by average pooling per channel, then normalization as follows:

$$P_l^r = \frac{1}{k} \sum_{i=1}^k \frac{1}{N} \sum_{x \in X} F^{ri}(x) Y_l^i(x), \quad (3.4a)$$

$$\hat{P}_l^r = \frac{P_l^r}{\|P_l^r\|_2}, \quad (3.4b)$$

here  $Y_l^i$  is a binary mask for  $i^{th}$  image with the novel class  $l$ ,  $F^{ri}$  is the corresponding output feature maps for  $i^{th}$  image and  $r^{th}$  resolution.  $X$  is the set of all possible spatial locations,  $k$  is the number of images in support set, and  $N$  is the number of pixels that are labelled as foreground for class  $l$ . The

normalized output from the masked average pooling layer  $\hat{P}_l^r$  can be further used as proxies representing class  $l$  and resolution  $r$ . In the case of a novel class the proxy can be utilized directly as the weight filter. An average of all the masked pooling features for the  $k$ -shot samples provided in the support set is used.

We denote this layer as a normalized masked average pooling as shown in Fig. 3.2. A similar layer is developed in a concurrent work [203]. It uses the output to compute a guidance to the network, while our method uses the output proxy to imprint the  $1 \times 1$  convolutional layer weights. This is the reason we use normalization. It is worth noting that the model is trained using a cosine similarity layer that performs normalization on both features and weights before predicting the segmentation logits.

### 3.2.5 Adaptive Proxies

We are inspired by the classical approaches in learning adaptive correlation filters [8], [58]. Correlation filters date back to 1970s [24], [59], [157]. More recently, the fast object tracking method MOSSE [8] relied on handcrafted features to form the correlation filters and adapted them using a running average. In our method the adaptation of the previously learned weights is based on a similar approach, yielding the ability to process base classes along with novel classes. It is valuable to utilize both instead of solely imprinting the new class weights. At the same time, in the case of the previously learned classes, e.g. background, it is not wise to simply override what the network learned from the large-scale training either. A good example illustrating the need to update the negative classes is the addition of “boat”. It is obvious that the *background* class needs to be updated to match the “sea” background, especially if the image with sea background are not part of the large scale training dataset.

We propose to update the convolutional layer weights in our model with the masked proxies for a given class using the following exponentially smoothed average adaptive scheme. As shown in Equation (3.5):

$$\hat{W}_l^r = \alpha \hat{P}_l^r + (1 - \alpha) W_l^r, \quad (3.5)$$

exponential smoothing is used to update the weights for older classes with the update rate  $\alpha$ .  $\hat{P}_l^r$  is the normalized masked proxy for class  $l$ ,  $W_l^r$  is the previously learned  $1 \times 1$  convolutional filter at resolution  $r$ , while  $\hat{W}_l^r$  is the adapted one. The update rate can either be treated as a hyper parameter or it can be learned separately according to the input embeddings. It can also be a learnable scalar value or it can vary according to which neuron is being updated. Fig. 3.2 shows our proposed adaptive masked proxies and its use to imprint the weights with each new support set. The new class weights are imprinted directly while the previously learned classes weights are updated. During the few-shot setup the support set contains segmentation masks for the new class (foreground) and base classes (background). Thus, the adaptation process is performed on all the base classes and then the output predictions are modified to be binary for background versus foreground (novel class). During inference, at the end of each sampled task once we imprint weights and infer on the query image we roll back the final layer weights to the ones used before imprinting. It is worth noting that recent work in unsupervised learning has also investigated the idea of adapting the old weights with new weights in a momentum encoder using an exponentially smoothed average in [55], which conforms with our proposed adaptation if the last layer’s weights are looked upon as prototypes that can be imprinted and adapted.

### 3.2.6 Multiresolution Imprinting Scheme

The adaptive masked proxies are used as the  $1 \times 1$  convolutional filters in a skip architecture [92]. The final classification layer, and the two  $1 \times 1$  convolutional layers following dilated convolutions in the case of Dilated-FCN8s are the ones imprinted. In case of FCN8s that does not utilize dilated convolution, the imprinted filters are used in the  $1 \times 1$  convolutional layers following the third and fourth pooling layers. For simplicity, an encoder-decoder architecture with skip connections was employed instead of pyramid processing. Additionally,

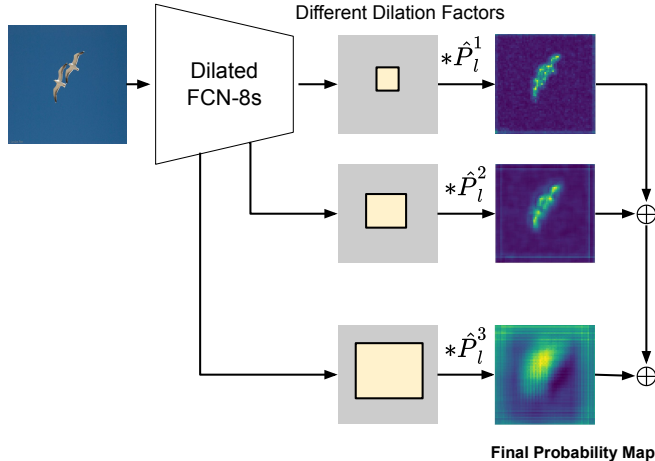


Figure 3.3: Multi-resolution imprinting using proxies from different resolution levels.

with atrous spatial pyramid pooling, large dilation rates as used in [19], results in large receptive fields that can affect the masked average pooling process. Fig. 3.3 shows the the output heatmaps from  $1 \times 1$  convolution using our proposed proxies as imprinted weights on three different resolution  $\hat{P}_l^1$ ,  $\hat{P}_l^2$ ,  $\hat{P}_l^3$ . It shows that the coarse resolution captures blobs necessary for global alignment, while the fine resolution provides the granular details required for an accurate segmentation.

To motivate why we have picked  $\hat{P}_l^r$  to act as proxies for the classes with few labelled samples, we plot the T-SNE [96] embedding for the learned proxies using normalized masked average pooling in Fig. 3.4. The plot shows the 5 classes belonging to fold 0 in Pascal-5<sup>i</sup>. Since our model performs imprinting on multiple resolution levels, the plot visualizes for the 3 different resolution levels. It also shows that better clustering happens in the intermediate layer, which confirms previous findings in a different problem setting in unsupervised learning [15].

### 3.3 Experimental Results

Our proposed method’s sample efficiency is evaluated on Pascal-5<sup>i</sup> and Labelled Faces in the Wild (LfW) [67]. In the few-shot segmentation scenario



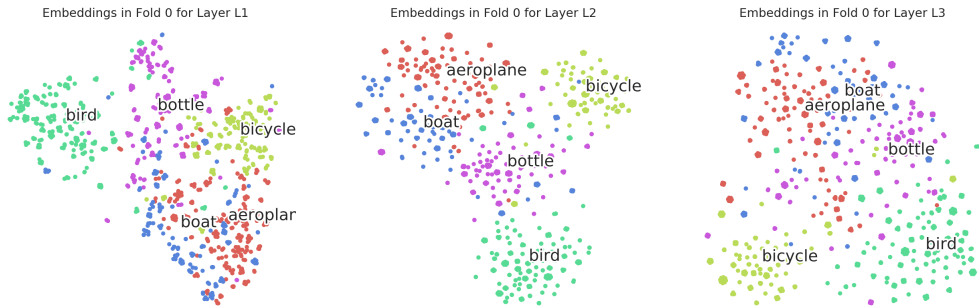


Figure 3.4: Visualization for the T-SNE [96] embeddings for the generated masked proxies for novel classes unseen during training. Layers L1, L2, L3 denote the smaller to higher resolution feature maps.

our method is evaluated on Pascal-5<sup>i</sup> [136]. An ablation study is performed to demonstrate the improvement from multi-resolution imprinting, and adaptive proxies. The study also compares weight imprinting against back-propagating on randomly generated weights. We use mean intersection over union (mIoU) similar to [136], where per-class IoU is computed and the mIoU denotes the average of the classes IoU per fold. Our code <sup>1</sup> is made publicly available to further benefit the few-shot learning research community. Our code base has built upon the semantic segmentation work [137], where we use Pytorch library [114].

### 3.3.1 Experimental Setup

The setup for pretraining the models to be tested on Pascal-5<sup>i</sup> is detailed. The base network is trained using RMSProp [60] with learning rate  $10^{-6}$ , and L2 regularization with a factor of  $5 \times 10^{-4}$  on the 15 classes outside of the current test fold for a fixed number of iterations 300,000. In the few-shot evaluation 1000 samples with support and query sets are used, similar to OLSM setup [136].

A hyper-parameter random search is conducted on the  $\alpha$  parameter, number of iterations, and the learning rate. The search is conducted on 10 classes from the training set and imprinting on the other 5 classes of the training set.

<sup>1</sup><https://github.com/MSiam/AdaptiveMaskedProxies>

Table 3.1: Quantitative results for 1-way 1-shot and 5-shot segmentation on Pascal-5<sup>i</sup> dataset following evaluation in [33]. FT: Fine-tuning for 10 iterations in 1-shot and 2 iterations in the 5-shot setting.

Method	1-Shot	5-Shot
FG-BG [33]	55.1	55.6
OSLSM [136]	55.2	-
co-FCN [121]	60.1	60.8
PL+SEG [33]	61.2	62.3
AMP-2 Norm (ours)	<b>62.3</b>	<b>66.6</b>

Thus ensuring all the classes used are outside the fold used in the test phase. The  $\alpha$  parameter selected is 0.26. In the case of performing fine-tuning, the selected learning rate is  $7.6 \times 10^{-5}$  with 2 iterations in the 5-shot case, and same learning rate with 5 iterations for the 1-shot case.

### 3.3.2 1-Way Few-Shot Semantic Segmentation

Table 3.2 and Table 3.3 show the results for the 1-shot and 5-shot segmentation respectively on Pascal-5<sup>i</sup> using mIoU of the foreground class. The 1-shot results differ than what is reported in the paper [147] as it does not incorporate the iterative refinement step, additionally feature normalization during training is added here. Our method is compared to OSLSM [136] and the baseline methods for few-shot segmentation. It shows that our method outperforms the baseline fine-tuning [136] method by 10.7% in terms of mIoU, without the need for extra back-propagation iterations through directly using the adaptive masked proxies. Our method out-performs OSLSM [136] method in the 1-shot and the 5-shot cases. However, unlike OSLSM our method does not need to train an extra branch for predicting the parameters.

Our method outperforms the co-FCN [121] method as shown in Table 3.2 by 2.2%. Fig. 3.5 shows the qualitative results on Pascal-5<sup>i</sup> which shows both the support set image-label pair, and our predicted segmentation for the query image. It shows that it does not depend on the saliency of the object. Since in some of the query images multiple potential objects can be categorized as salient, but it rather learns to segment what best matches the proxy.

Table 3.1 shows our method in comparison to the state of the art methods

Table 3.2: Quantitative results for 1-way 1-shot segmentation on Pascal-5<sup>i</sup> dataset. FT: Fine-tuning. AMP-1: using Dilated FCN8s. AMP-2: using Reduced version of Dilated FCN8s. AMP-2 Norm: Similar to AMP-2 with using cosine similarity layer with normalization on features and weights during the training phase not only inference. Red, Blue: Best and Second Best Performing Methods. Co-FCN evaluation using Fg only reported from [203]. Bolded numbers indicate best performance.

	Fold 1	Fold 2	Fold 3	Fold 4	Mean-IoU
1-NN [136]	25.3	44.9	41.7	18.4	32.6
Siamese [136]	28.1	39.9	31.8	25.8	31.4
FT [136]	24.9	38.8	36.5	30.1	32.6
OSLSM [136]	33.6	<b>55.3</b>	40.9	33.5	40.8
Co-FCN [121]	36.7	50.6	44.9	32.4	41.1
AMP-1 (ours)	37.4	50.9	46.5	34.8	42.4
AMP-2 (ours)	38.5	52.0	46.6	<b>36.0</b>	<b>43.3</b>
AMP-2 Norm (ours)	<b>39.6</b>	52.1	<b>46.7</b>	34.6	<b>43.3</b>

Table 3.3: Quantitative results for 1-way 5-shot segmentation on Pascal-5<sup>i</sup> dataset. AMP-2 + FT(2): fine-tuning with 2 iterations after our proposed method. Red, Blue: Best and Second Best Performing Methods. Co-FCN evaluation using Fg only reported from [203]. Bolded numbers indicate best performance.

	Fold 1	Fold 2	Fold 3	Fold 4	Mean-IoU
1-NN [136]	34.5	53.0	46.9	25.6	40.0
LogReg [136]	35.9	51.6	44.5	25.6	39.3
OSLSM [136]	35.9	<b>58.1</b>	42.7	39.1	43.9
co-FCN [121]	37.5	50.0	44.1	33.9	41.4
AMP-2 (ours)	40.3	55.3	49.9	40.1	46.4
AMP-2 + FT(2) (ours)	41.8	55.5	50.3	39.9	46.9
AMP-2 Norm (ours)	<b>44.5</b>	57.3	<b>50.8</b>	<b>41.4</b>	<b>48.5</b>

in terms of the mean on all folds for 1-shot and 5-shot segmentation with a different evaluation. The same evaluation utilized by [121] and [33] is used, which computes mIoU as the mean of the foreground and background IoU. Our proposed method outperforms state of the art methods in both the 1-shot and 5-shot cases. It shows a larger improvement in case of the 5-shot than the 1-shot, since the 5-shot averages the prototype from the 5 images, which results in a better class signature.

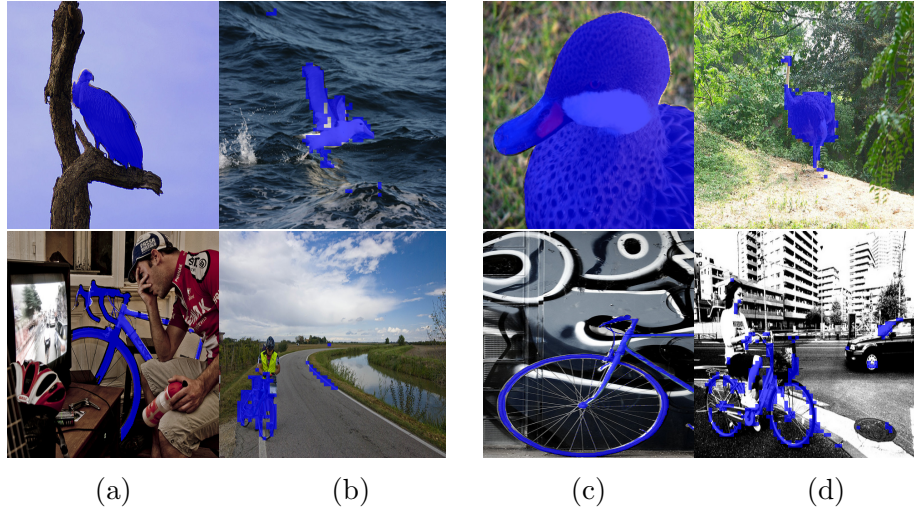


Figure 3.5: Qualitative evaluation on Pascal-5<sup>i</sup> 1-way 1-shot. The support set (a,c) and prediction (b,d) on the query image are shown.

### 3.3.3 2-Way Few-Shot Semantic Segmentation

Experiments conducted on Pascal-5<sup>i</sup> similar to Shaban et. al. [136] setup evaluate 1-way segmentation with the new class as the foreground to be predicted. We conduct further experiments on Labelled Faces in the Wild (LFW) dataset [67] to perform 2-way segmentation. The dataset provides images for labelled faces annotated with parts which include two semantic part classes. We compare against initializing the weights of the convolutional layer responsible for classification randomly, and with naive fine-tuning on the random weights. Evaluation is performed on the 1-shot and 5-shot cases as shown in Table 3.4. Our proposed method outperforms the naive finetuning baseline with a significant margin of 24.6% and 23.4% in the 1-shot case and 5-shot case respectively. Performing fine-tuning with our masked proxies scheme leads to a boost in the mIoU with 3.0% in the 1-shot and 4.4% in the 5-shot. The flexibility of the proposed few-shot method allows its coupling with back-propagation which proves to be useful in the parts segmentation. Nonetheless, using the masked proxies solely without fine-tuning outperforms naive fine-tuning with randomly initialized weights. Figure 3.6 shows qualitative evaluation on LFW dataset for the 1-shot 2-way segmentation scenario.

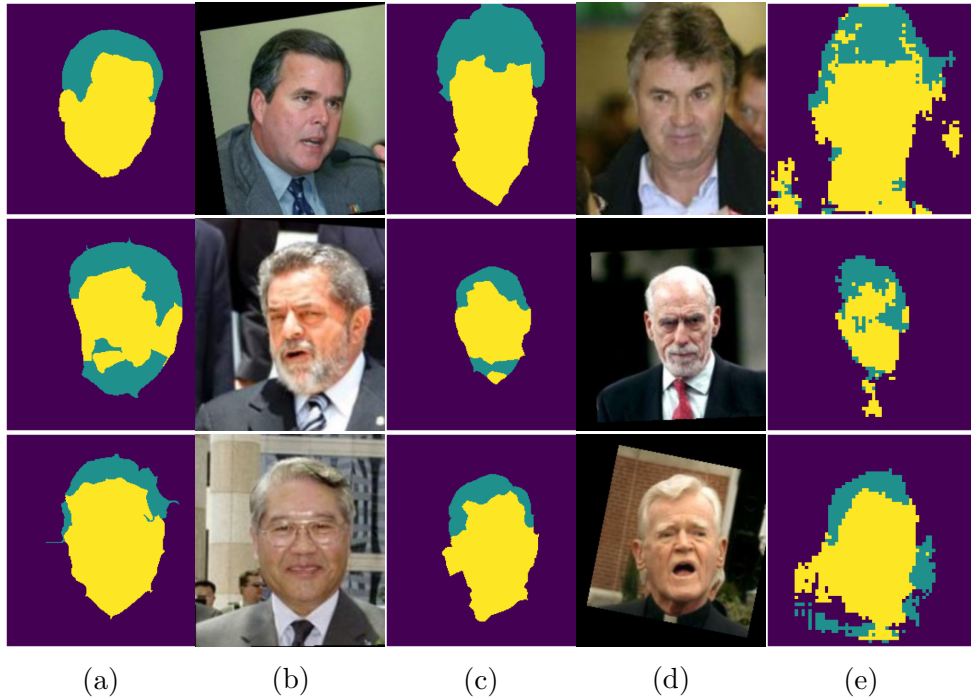


Figure 3.6: Output predictions on LFW dataset [67] using our proposed method without fine-tuning. (a,b) support set label-image pair. (c,d) query label-image pair. (e) our prediction.

Table 3.4: Quantitative results on LFW [67] segmentation dataset. Rnd: Indicates the use of random weights in the final layer. MP: indicates imprinting the final layer weights using the masked proxies. FT: indicates fine-tuning. Bolded numbers indicate best performance.

Method	1-Shot	5-shot
Rnd	15.0	-
MP (ours)	48.4	53.8
Rnd + FT	20.4	27.5
MP + FT (ours)	<b>50.2</b>	<b>58.7</b>

### 3.3.4 Ablation Study

We perform an ablation study on fold 0 to ensure the effectiveness of different components in our proposed method. Table 3.5 shows the benefit from our proposed method, it outperforms fine-tuning using randomly generated weights with a significant margin. It also shows the benefit from proposing an adaptive method, where no adaptation with  $\alpha$  set to 0, degrades accuracy. It demonstrates that directly imprinting the weights for the new class solely is

Table 3.5: Ablation study of the different design choices for the imprinting scheme on Fold 0. Adaptation:  $\alpha$  parameter is non-zero. Multi-res: performing multi-resolution imprinting. Imp: imprinting weights using our proxies. FT: fine-tuning. Norm: Normalization.

Methods	Adap.	Multires.	Norm.	N-Shot	mIoU
Ft Only	✗	✓	✗	5	28.7
Imp.	✓	✓	✓	5	<b>44.5</b>
CosSim + Ft. Only	✗	✓	✓	1	10.6
Imp.	✗	✓	✗	1	13.6
Imp.	✓	✗	✗	1	34.8
Imp.	✓	✓	✗	1	37.4
Imp.	✓	✓	✓	1	<b>39.6</b>

Table 3.6: Evaluating mean IoU over 15 base classes along with the extra novel class over 4 folds on Pascal-5i. Comparing our two variants based on the model trained with Cosine Similarity layer when fine-tuning with random weights or imprinting the weights and performing adaptation.

Method	N-shot	Fold 0	Fold 1	Fold 2	Fold 3	mIoU
CosSim + Imp. (ours)	1-shot	12.2	11.4	8.3	11.7	10.9
CosSim + Ft.	1-shot	12.6	11.9	12.2	15.2	13.0

not sufficient and has to be coupled with our proposed adaptive scheme. We also motivate the use of our proposed multi-resolution imprinting. As shown in Table 3.5 it outperforms the method that does not support multi-resolution. Results in Table 3.5 for our final method corresponds to the evaluation method and results provided in Table 3.2 and Table 3.3 on fold 0 following Shaban et. al. [136]. Finally, it shows that using cosine similarity layer during training in which both features and weights are normalized before performing the convolution followed by scaling can improve the results further on.

We further report the mean IoU computed over all 15 base classes and the novel class after performing the weight imprinting on the 1-shot case in Table 3.6. We compare our two variants trained with a cosine similarity layer when both finetuning or imprinting the weights directly and performing adaptation. In the case of generalized few-shot setting the fine-tuning performs better than imprinting then adaptation. Although in Table 3.5 when evaluating 1-way the imprinting mechanism with adaptation is better. It indicates

that there is still work that can be done to improve the results on the generalized few-shot segmentation setting, and ensuring that the adaptation does not result in decreased performance on base classes. Future work on learning the alpha parameter used in adaptation and varying it from one class to another would be a good possible direction to improve the generalized few-shot setting results. Overall, the results demonstrate that imprinting and adaptation mechanism provide a way to improve the accuracy in the 1-way segmentation.

### 3.4 Summary

In this chapter we proposed a sample efficient method to segment unseen classes via multi-resolution imprinting of adaptive masked proxies (AMP). AMP constructs the final segmentation layer weights from few labelled support set samples by imprinting the masked multi-resolution responses of the base feature extractor and by fusing it with the previously learned class signatures. AMP is empirically validated to be superior in the few-shot segmentation on PASCAL-5i benchmark with a significant margin in the 5-shot case. Additionally, experiments for 2-way on labelled Faces in the Wild dataset shows the advantage from using such approach over simple fine-tuning of randomly generated weights. However, the adaptation mechanism shows weaknesses in the generalized few-shot segmentation setup which we leave for future work.

# Chapter 4

## Few-shot Weakly Supervised Object Segmentation using Co-Attention

### 4.1 Introduction

Existing literature in few-shot object segmentation has mainly relied on manually labelled segmentation masks including our previous method. A few recent works [121], [177], [201] started to conduct experiments using weak annotations such as scribbles or bounding boxes. However, these weak forms of supervision involve more manual work compared to image level labels, which can be collected from text and images publicly available on the web. Little research has been conducted on using image-level supervision for few-shot segmentation [125]. Performance of most of the current weakly supervised few-shot segmentation methods lag significantly behind their strongly supervised counterparts.

Few-shot object segmentation literature at the time of this work was mainly focused on using a single vector representation from masked average pooling to guide the query image segmentation. However, single vector representation would lose critical information required for detailed object segmentation, on top of that with image-level labels it is not possible to perform masked average pooling. If a global average pooling would be used, this would lead to the confusion of foreground and background features. In this work we propose to leverage the interaction between the support set and query image using



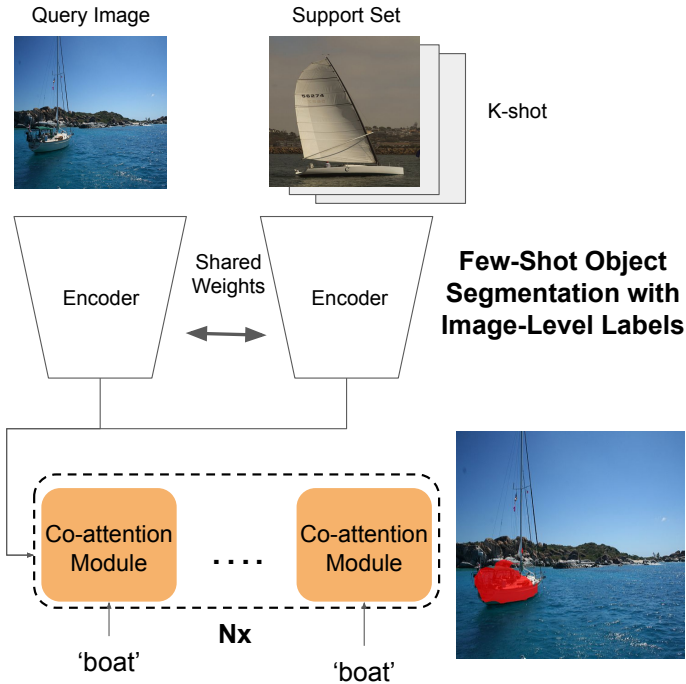


Figure 4.1: Overview of stacked co-attention to relate the support set and query image using image-level labels.  $Nx$ : Co-attention stacked  $N$  times. “K-shot” refers to using  $K$  support images.

co-attention, which is based on pixel-to-pixel affinity matrices. Furthermore, to improve the weakly supervised few-shot object segmentation method we propose to utilize the semantic word embeddings as a conditioning signal to the co-attention module as shown in Fig. 4.1, which we call a multi-modal interaction module. Our method outperforms [125] by 4.8% and improves over methods that use bounding box supervision [177], [201].

Most work in few-shot segmentation considers the *static* setting where query and support images do not have temporal relations. However, in real world applications such as robotics, segmentation methods can benefit from temporal continuity and multiple viewpoints. For real time segmentation, it may be of tremendous benefits to utilize temporal knowledge existing in video sequences. Observations that pixels moving together mostly belong to the same object seem to be very common in videos, and it can be exploited to improve segmentation accuracy. This is referred to as the Gestalt’s notion of

“Common Fate” [170]. We propose a novel setup, temporal object segmentation with few-shot learning (TOSFL), where support and query images are temporally related. The TOSFL setup for video object segmentation generalizes to novel object classes as can be seen in our experiments on Youtube-VOS dataset [194]. TOSFL only requires image-level labels for the first frames (support images) to segment the objects that appear in the frames that follow. The TOSFL setup is interesting because it is more similar to the nature of learning of objects by a human [170] than the strongly supervised static segmentation setup.

Youtube-VOS [194] provides a way to evaluate on unseen categories. However, it does not utilize the category labels in the segmentation model. Our setup relies on the image-level label for the support image to segment different parts from the query image conditioned on the word embeddings of this image-level label. In order to ensure the evaluation for the few-shot method is not biased to a certain category, it is best to split into multiple folds and evaluate on different ones similar to [136].

**Contributions**

- We propose a novel few-shot object segmentation algorithm based on a multi-stage co-attention mechanism that leverages the interaction between support and query features, presented in Sections 4.2.1 and 4.2.2.
- We propose semantic conditioning to alleviate ambiguity caused by confusion from base classes or multiple common objects between support and query images, discussed in Section 4.2.1.
- We further propose a novel weakly supervised few-shot video object segmentation setup, as detailed in Section 4.2.3. It complements the existing few-shot object segmentation benchmarks by considering a practically important use case not covered by previous datasets. Video sequences are provided instead of static images which can simplify the few-shot learning problem.

### 4.1.1 Attention Mechanisms

Since the current work mostly focuses on exploring co-attention between support and query frames we gather necessary related work on the different attention mechanisms. Attention was initially proposed for neural machine translation models [4]. Several approaches were proposed for utilizing attention. [197] proposed a stacked attention network which learns attention maps sequentially on different levels. [93] proposed co-attention to solve a visual question and answering task by alternately shifting attention between visual and question representations. [94] used co-attention in video object segmentation between frames sampled from a video sequence. Concurrent to our work, new work on using co-attention for one-shot object detection was proposed in [63]. However, they mainly use it to attend to the query image since the given bounding box provides them with the region of interest in the support set image. To the best of our knowledge, this work is the first one to explore the bidirectional attention between support and query sets in an iterative manner as a mechanism for solving the few-shot image segmentation task with image-level supervision.

A recent work exploring self attention for image recognition has defined a unifying framework for attention where it is divided into pairwise and patch-wise attention. Pairwise attention follows the formulation in Equation 4.1:

$$y_i = \sum_{j \in R(i)} \alpha(x_i, x_j) \circ \beta(x_j), \tag{4.1}$$
$$\alpha(x_i, x_j) = \gamma(\delta(x_i, x_j)),$$

where  $\gamma$  and  $\beta$  are nonlinear projection functions, and  $R(i)$  is the local footprint used for aggregation. While the  $\delta$  relation function can either be hadamarad product, summation, subtraction, or dot product. On the other hand, patch-wise attention considers local connections in the relation function. As shown in Equation 4.2:

$$\begin{aligned}
y_i &= \sum_{j \in R(i)} \alpha(x_{R(i)})_j \circ \beta(x_j), \\
\alpha(x_{R(i)}) &= \gamma(\delta(x_{R(i)})),
\end{aligned} \tag{4.2}$$

$$\delta(x_{R(i)}) = [\phi(x_i), [\psi(x_j)]_{\forall j \in R(i)}],$$

the  $\delta$  function can either be star-product, clique-product or concatenation. Concatenation is the one used in Equation 4.2 as an example. Our method for performing co-attention follows a pairwise formulation where  $\delta$  is a simple dot product. We leave for future work exploring local connections with patchwise attention.

## 4.2 Proposed Method

The human perception system is inherently multi-modal. Inspired from this and to leverage the learning of new concepts we propose a multi-modal interaction module that embeds semantic conditioning in the visual processing scheme as shown in Fig. 4.2. The overall model consists of: (1) Encoder. (2) Multi-modal Interaction module. (3) Segmentation Decoder. The multi-modal interaction module is described in detail in this section while the encoder and decoder modules are explained in Section 4.3.1. We follow a 1-way  $k$ -shot setting similar to [136].

### 4.2.1 Multi-Modal Interaction Module

One of the main challenges in dealing with the image-level annotation in few-shot segmentation is that quite often both support and query images may contain a few salient common objects from different classes. Inferring a good prototype for the object of interest from multi-object support images without relying on pixel-level cues or even bounding boxes becomes particularly challenging. Yet, it is exactly in this situation, that we can expect the semantic word embeddings to be useful at helping to disambiguate the object relationships across support and query images. Below we discuss the technical details behind the implementation of this idea depicted in Fig. 4.2. Initially, in a  $k$ -shot setting, a base network is used to extract features from  $i^{th}$  support set

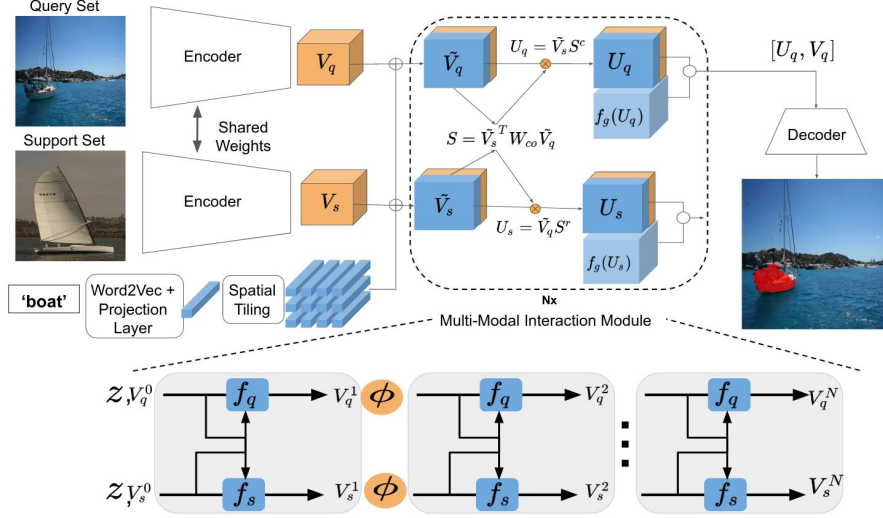


Figure 4.2: Architecture of Few-Shot Object segmentation model with co-attention. The  $\oplus$  operator denotes concatenation,  $\circ$  denotes element-wise multiplication. Only the decoder and multi-modal interaction module parameters are learned, while the encoder is pretrained on ImageNet. The stacked co-attention iterations are unrolled for visualisation solely.

image  $I_s^i$  and from the query image  $I_q$ , which we denote as  $V_s \in R^{W \times H \times C'}$  and  $V_q \in R^{W \times H \times C'}$ . Here  $H$  and  $W$  denote the height and width of feature maps, respectively, while  $C'$  denotes the number of feature channels. Furthermore, a projection layer is used on the semantic word embeddings to construct  $z \in R^d$  ( $d = 256$ ). It is then spatially tiled and concatenated with the visual features resulting in flattened matrix representations  $\tilde{V}_q \in R^{C \times WH}$  and  $\tilde{V}_s \in R^{C \times WH}$ . An affinity matrix  $S$  is computed to capture the similarity between them via a fully connected layer  $W_{co} \in R^{C \times C}$  learning the correlation between feature channels:

$$S = \tilde{V}_s^T W_{co} \tilde{V}_q.$$

the affinity matrix  $S \in R^{WH \times WH}$  relates each pixel in  $\tilde{V}_q$  and  $\tilde{V}_s$ . A softmax operation is performed on  $S$  row-wise and column-wise depending on the desired direction of relation:

$$S^c = \text{softmax}(S), \quad S^r = \text{softmax}(S^T)$$

For example, column  $S_{*,j}^c$  contains the relevance of the  $j^{th}$  spatial location in

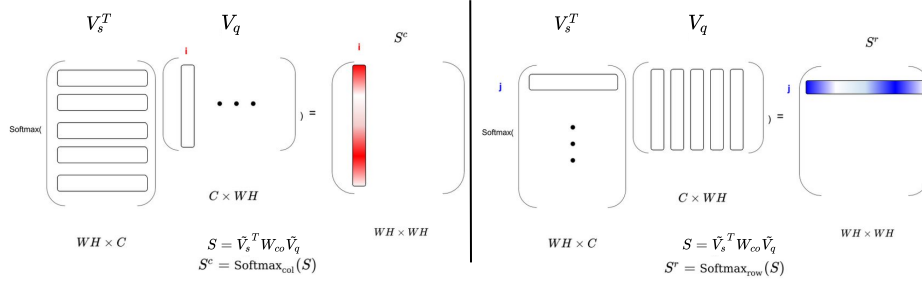


Figure 4.3: Visual explanation of pixel-to-pixel affinity matrix used to compute attention weights.

$V_q$  with respect to all spatial locations of  $V_s$ , where  $j = 1, \dots, WH$  as shown in Figure 4.3. The normalized affinity matrix is used to compute attention summaries  $U_q$  and  $U_s$  through reweighting it with the attention weights from  $S^c$  or  $S^r$  respectively:

$$U_q = \tilde{V}_s S^c, \quad U_s = \tilde{V}_q S^r.$$

The attention summaries are further reshaped such that  $U_q, U_s \in R^{W \times H \times C}$  and gated using a gating function  $f_g$  with learnable weights  $W_g$  and bias  $b_g$ :

$$f_g(U_q) = \sigma(W_g * U_q + b_g),$$

$$U_q = f_g(U_q) \circ U_q.$$

Here the  $\circ$  operator denotes element-wise multiplication. The gating function restrains the output to the interval  $[0, 1]$  using a sigmoid activation function  $\sigma$  in order to mask the attention summaries. The gated attention summaries  $U_q$  are concatenated with the original visual features  $V_q$  to construct the final output from the attention module to the decoder.

## 4.2.2 Stacked Gated Co-Attention

We propose to stack the multi-modal interaction module described in Section 4.2.1 to learn an improved representation. Stacking allows for multiple

Table 4.1: Notations Summary Table.

Symbol	Definition
$V_q, V_s$	Query, Support visual features.
$\tilde{V}_q$	Query visual and word embeddings concatenated.
$z$	class label word embeddings.
$U_q, U_s$	Query, support attention summaries.
$W_{co}$	Weight matrix in Co-Attention.
$L_{train}$	Set of training classes.
$D_{train}$	Set of training data.
$H, W$	Height and Width of the visual features.
$C$	Channels of the visual and semantic features combined.
$S$	Affinity matrix.
$f_g$	Gating function.
$W_g, b_g$	Weight and bias used in the gating function.
$f_q$	Function to compute query attention summaries.

iterations between the support and the query images. The co-attention module has two streams  $f_q, f_s$  that are responsible for processing the query image and the support set images respectively. The inputs to the co-attention module,  $V_q^i$  and  $V_s^i$ , represent the visual features at iteration  $i$  for query image and support image respectively. In the first iteration,  $V_q^0$  and  $V_s^0$  are the output visual features from the encoder. Each multi-modal interaction then follows the recursion  $\forall i = 0, \dots, N - 1$ :

$$V_q^{i+1} = \phi(V_q^i + f_q(V_q^i, V_s^i, z))$$

The nonlinear projection  $\phi$  is performed on the output from each iteration, which is composed of  $1 \times 1$  convolutional layer followed by a ReLU activation function. We use residual connections in order to improve the gradient flow and prevent vanishing gradients. The support set features  $V_s^i, \forall i = 0, \dots, N - 1$  are computed similarly. We illustrate the stacked co-attention in Figure 4.2. Refer to Table 4.1 to summarize notations used in the paper.

### 4.2.3 Temporal Object Segmentation with a Few-shot Learning Setup

We propose a novel few-shot video object segmentation (VOS) task. In this task, the image-level label of the first frame is provided to learn object seg-

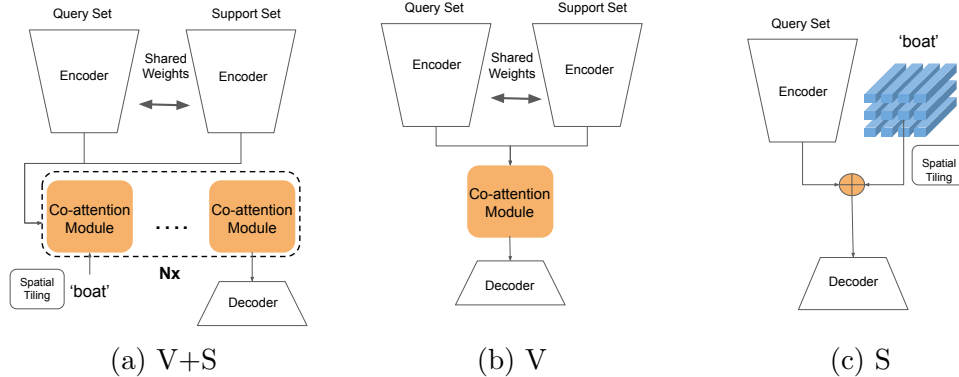


Figure 4.4: Different variants for image-level labelled few-shot object segmentation. V+S: Stacked Co-Attention with Visual and Semantic representations. V: Co-Attention with Visual features only. S: Conditioning on semantic representation only from word embeddings.

mentation in the sampled frames from the ensuing sequence. This is a more challenging task than the one relying on the pixel-level supervision in semi-supervised VOS. The task is designed as a binary segmentation problem and the categories are split in multiple folds, consistent with existing few-shot segmentation tasks defined on Pascal-5<sup>i</sup> and MS-COCO. This design ensures that the proposed task assesses the ability of few-shot video segmentation algorithms to generalize over unseen classes. We utilize Youtube-VOS dataset training data which has 65 classes, and we split them into 5 folds. Each fold has 13 classes that are used as novel classes, while the rest are used in the meta-training phase. A randomly sampled class  $Y_s$  and sequence  $V = \{I_1, I_2, \dots, I_N\}$  are used to construct the support set  $S_p = \{(I_1, Y_s)\}$  and query images  $I_i$ . For each query image a binary segmentation mask  $M_s^Y$  is constructed by labelling all the instances belonging to  $Y_s$  as foreground. Accordingly, the same image can have multiple binary segmentation masks depending on the sampled  $Y_s$ .

### 4.3 Experimental Results

In this section we demonstrate results of experiments conducted on the Pascal-5<sup>i</sup> dataset [136] compared to state of the art methods in Section 4.3.2. Not only do we set strong baselines for image level labelled few shot segmentation and outperform previously proposed work [125], but we also perform close



to the state of the art conventional few shot segmentation methods that use detailed pixel-wise segmentation masks. We then demonstrate the results for the different variants of our approach depicted in Fig. 4.4 and experiment with the proposed TOSFL setup in Section 4.3.3.

### 4.3.1 Experimental Setup

**Network Details:** We utilize a ResNet-50 [57] encoder pre-trained on ImageNet [31] to extract visual features. The segmentation decoder is comprised of an iterative optimization module (IOM) [201] and an atrous spatial pyramid pooling (ASPP) [19], [20]. The IOM module takes the output feature maps from the multi-modal interaction module and the previously predicted probability map in a residual form. The previously predicted probability maps are initially set to zeros and then are set to the output from the previous iteration. Following the IOM module, an ASPP module is used to increase the model receptive field and to capture the global context following the modification from DeepLab-V3 [20].

**Meta-Learning Setup:** We sample 12,000 tasks during the meta-training stage. In order to evaluate test performance, we average accuracy over 5000 tasks with support and query sets sampled from the meta-test dataset  $D_{test}$  belonging to classes  $L_{test}$ . We perform 5 training runs with different random generator seeds and report the average of the 5 runs and the 95% confidence interval.

**Evaluation Protocol:** Pascal-5<sup>i</sup> splits PASCAL-VOC 20 classes into 4 folds each having 5 classes. The mean IoU and binary IoU are the two metrics used for the evaluation process. The mIoU computes the intersection over union for all 5 classes within the fold and averages them neglecting the background. Whereas the bIoU metric proposed by [121] computes the mean of foreground and background IoU in a class agnostic manner. We have noticed some deviation in the validation schemes used in previous works. Zhang et. al. [201] follow a procedure where the validation is performed on the test classes to save the best model, whereas Wang et. al. [177] rather trains for a fixed number of iterations for all models. We choose the more challenging approach

in [177].

**Training Details:** During the meta-training, we freeze ResNet-50 encoder weights while learning both the multi-modal interaction module and the decoder. We train all models using momentum SGD with learning rate 0.01 that is reduced by 0.1 at epoch 35, 40 and 45 and momentum 0.9. L2 regularization with a factor of  $5 \times 10^{-4}$  is used to avoid over-fitting. Batch size of 4 and input resolution of  $321 \times 321$  are used during training with random horizontal flipping and random centered cropping for the support set. An input resolution of  $500 \times 500$  is used for the meta-testing phase similar to [136]. In each fold the model is meta-trained for a maximum number of 50 epochs on the classes outside the test fold.

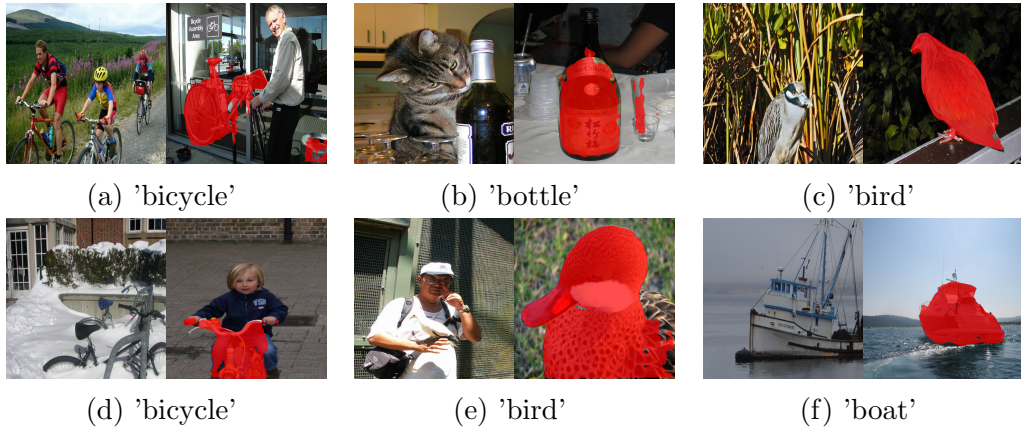


Figure 4.5: Qualitative evaluation on Pascal-5<sup>i</sup> 1-way 1-shot. The support set and prediction on the query image are shown in pairs.



Figure 4.6: Failure cases on Pascal-5<sup>i</sup> 1-way 1-shot. The support set and prediction on the query image are shown in pairs.

Table 4.2: Quantitative results for 1-way, 1-shot segmentation on the Pascal-5<sup>i</sup> dataset showing mean-Iou and binary-IoU. P: stands for using pixel-wise segmentation masks for supervision. IL: stands for using weak supervision from Image-Level labels. BB: stands for using bounding boxes for weak supervision. Red: validation following [201], Blue: validation following [177].

Method	Type	1-shot					
		1	2	3	4	mIoU	bIoU
FG-BG	P	-	-	-	-	-	55.1
OSLSM [136]	P	33.6	55.3	40.9	33.5	40.8	-
CoFCN [121]	P	36.7	50.6	44.9	32.4	41.1	60.1
PLSeg [33]	P	-	-	-	-	-	61.2
AMP [147]	P	41.9	50.2	46.7	34.7	43.4	62.2
PANet [177]	P	42.3	58.0	51.1	41.2	48.1	66.5
CANet [201]	P	52.5	65.9	51.3	51.9	55.4	66.2
PGNet [200]	P	56.0	66.9	50.6	50.4	56.0	69.9
CANet [201]	BB	-	-	-	-	<b>52.0</b>	-
PANet [177]	BB	-	-	-	-	<b>45.1</b>	-
OSW [125]	IL	-	-	-	-	-	<b>58.7</b>
Ours(V+S)-1	IL	49.5	65.5	50.0	49.2	<b>53.5</b>	<b>65.6</b>
Ours(V+S)-2	IL	42.5	64.8	48.1	46.5	<b>50.5</b>	<b>64.1</b>
						$\pm 0.7$	$\pm 0.4$

### 4.3.2 Comparison to the state-of-the-art

We compare the result of our best variant (see Fig. 4.4), *i.e.*: Stacked Co-Attention (V+S) against the other state of the art methods for 1-way 1-shot and 5-shot segmentation on Pascal-5<sup>i</sup> in Table 4.2 and 4.3. We report the results for different validation schemes. Ours(V+S)-1 follows [201] and Ours(V+S)-2 follows [177]. Without the utilization of segmentation mask or even sparse annotations, our method with the least supervision of image level labels performs (53.5%) close to the current state of the art strongly supervised methods (56.0%) in 1-shot case and outperforms the ones that use bounding box annotations. It improves over the previously proposed image-level supervised method with a significant margin (4.8%). For the  $k$ -shot extension of our method we perform average of the attention summaries during the meta-training on the  $k$ -shot samples from the support set. Table 4.4 demonstrates results on MS-COCO [87] compared to the state of the art method using pixel-wise segmentation masks for the support set.

Table 4.3: Quantitative results for 1-way, 5-shot segmentation on the Pascal-5<sup>i</sup> dataset showing mean-Iou and binary-IoU. P: stands for using pixel-wise segmentation masks for supervision. IL: stands for using weak supervision from Image-Level labels. BB: stands for using bounding boxes for weak supervision. Blue: validation following [177].

Method	Type	5-shot				
		1	2	3	4	mIoU
OSLSM [136]	P	35.9	58.1	42.7	39.1	43.9
CoFCN [121]	P	37.5	50.0	44.1	33.9	41.4
AMP [147]	P	41.8	55.5	50.3	39.9	46.9
PANet [177]	P	51.8	64.6	59.8	46.5	55.7
CANet [201]	P	55.5	67.8	51.9	53.2	57.1
PGNet [200]	P	57.7	68.7	52.9	54.6	58.5
PANet [177]	BB	-	-	-	-	<b>52.8</b>
Ours(V+S)-2	IL	45.9	65.7	48.6	46.6	<b>51.7</b> $\pm 0.07$

Table 4.4: Quantitative Results on MS-COCO Few-shot 1-way.

Method	Type	1-shot	5-shot
PANet [177]	P	20.9	29.7
Ours-(V+S)	IL	15.0	15.6

### 4.3.3 Ablation Study

We perform an ablation study to evaluate different variants of our method depicted in Fig. 4.4. Table 4.6 shows the results on the three variants we proposed on Pascal-5<sup>i</sup>. It clearly shows that using the visual features only (V-method), lags 5% behind utilizing word embeddings in the 1-shot case. This is mainly due to having multiple common objects between the support set and the query image, and tendency to segment objects from base classes. Semantic conditioning obviously helps to resolve the ambiguity and improves the result significantly as shown in Figure 4.7. Going from 1 to 5 shots, the V-method improves, because multiple shots are likely to repeatedly contain the object of interest and the associated ambiguity decreases, but still it lags behind both variants supported by semantic input. Interestingly, our results show that the baseline of conditioning on semantic representation is a very competitive variant: in the 1-shot case it even outperforms the (V+S) variant. However,

Table 4.5: Ablation Study for different components with 1 run on Pascal-5<sup>i</sup> and Youtube-VOS. V: visual, S: semantic. SCoAtt: Stack Co-Attention. Cond: Concatenation based conditioning.

Dataset	Method	mIoU
Pascal-5 <sup>i</sup>	V-Cond	42.7
Pascal-5 <sup>i</sup>	V-CoAtt	44.6
Pascal-5 <sup>i</sup>	V+S-Cond	50.1
Pascal-5 <sup>i</sup>	V+S-CoAtt	50.2
Pascal-5 <sup>i</sup>	V+S-SCoAtt	<b>51.0</b>
Youtube-VOS	V+S-Cond	42.3
Youtube-VOS	V+S-SCoAtt	<b>43.7</b>

Table 4.6: Ablation Study on 4 folds of Pascal-5<sup>i</sup> for few-shot segmentation for different variants showing mean-IoU. V: visual, S: semantic. V+S: both features.

Method	1-shot	5-shot
V	44.4 ± 0.3	49.1 ± 0.3
S	<b>51.2 ± 0.6</b>	51.4 ± 0.3
V+S	50.5 ± 0.7	<b>51.7 ± 0.07</b>

Table 4.7: Quantitative Results on Youtube-VOS One-shot weakly supervised setup showing IoU per fold and mean-IoU over all folds similar to Pascal-5<sup>i</sup>. V: visual, S: semantic. V+S: both features.

Method	1	2	3	4	5	Mean-IoU
V	40.8	34.0	44.4	35.0	35.5	38.0 ± 0.7
S	42.7	40.8	48.7	38.8	37.6	41.7 ± 0.7
V+S	<b>46.1</b>	<b>42.0</b>	<b>50.7</b>	<b>41.2</b>	<b>39.2</b>	<b>43.8 ± 0.5</b>

on the TOSFL setup the V+S variant shows significant improvement over the S-variant.

We perform an ablation study to evaluate different components of our method. Table 4.5 show results for 1 run and compares using a simple conditioning on the support set features through concatenation with the query visual features against performing co-attention between support and query feature maps. It shows clearly the benefit from performing co-attention. Nonetheless, visual features solely is not capable to disambiguate between different common objects and the visual with semantic embeddings even with simple concatenation shows an improvement. Further stacking the co-attention module proves

to improve the results as well specifically on Youtube-VOS.

Table 4.7 shows the results on our proposed novel video segmentation task, comparing variants of the proposed approach. As previously, the baseline V-method based on co-attention module with no word embeddings, similar to [94], lags behind both S- and (V+S)-methods. It is worth noting that unlike the conventional video object segmentation setups, the proposed video object segmentation task poses the problem as a binary segmentation task conditioned on the image-level label. We demonstrate in Table 4.7 that the (V+S)-method’s joint visual and semantic processing in such scenario clearly provides significant gain. The gain is mostly attributed to reducing the tendency to segment base classes used during the meta-training phase which was a problem in coattention without semantic conditioning in both Pascal-5i and Youtube-VOS. However, there is still tendency to segment the salient objects over small and not visually salient ones in Youtube-VOS that we are still looking into improving in our future work.

## 4.4 Summary

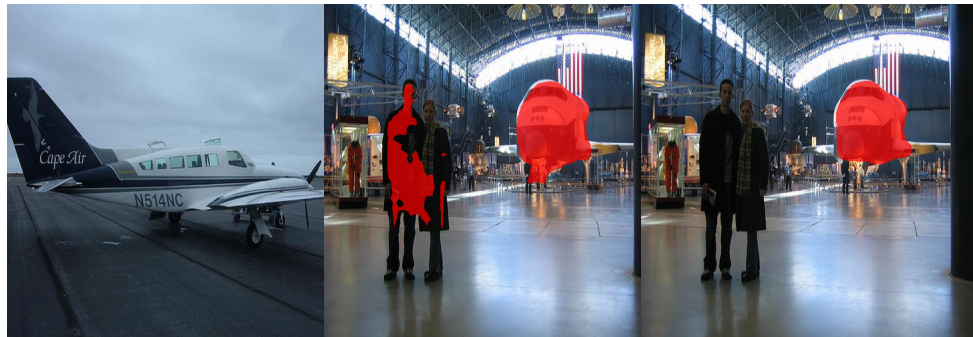
In this chapter we proposed a multi-modal interaction module that relates the support set image and query image using both visual and word embeddings. We proposed to meta-learn a stacked co-attention module that guides the segmentation of the query based on the support set features and vice versa. The two main takeaways from the experiments are that: (i) few-shot segmentation significantly benefits from utilizing word embeddings and (ii) it is viable to perform high quality few-shot segmentation using stacked joint visual semantic processing with weak image-level labels. Additionally, a temporal object segmentation for few-shot learning setup is proposed that bridges the gap between few-shot and video-object segmentation.



(a) Label 'bicycle'

(b) Prediction (V)

(c) Prediction (V+S)



(d) Label 'aeroplane'

(e) Prediction (V)

(f) Prediction (V+S)



(g) Label 'bird'

(h) Prediction (V)

(i) Prediction (V+S)

Figure 4.7: Qualitative analysis on fold 0 Pascal-5<sup>i</sup> between our method (V+S) and object co-segmentation baseline ours (V) that can not disambiguate multiple common objects and is biased toward base classes used in training.

**Part II**

**Video Object Segmentation  
(VOS)**



# Chapter 5

## Video Class Agnostic Segmentation

### 5.1 Introduction

Semantic scene understanding is crucial in autonomous driving in both end-to-end and mediated perception approaches. Semantic segmentation [25][104], which performs pixel-wise classification of the scene, and Panoptic segmentation [193][75][61], which combines semantic and instance segmentation, are both focused on a closed set of known classes. However, a system trained on a limited set of classes would face difficulties in unexpected situations that could occur in different autonomous driving scenarios as shown in Figure 5.1. The parking and construction scenarios we have built in CARLA are clear examples that provide multiple objects outside the closed set of known classes in most publicly available datasets.

In this chapter we formulate the task of video class agnostic segmentation in order to allow for the segmentation of unknown objects and publicly release the necessary datasets and baselines for the different task formulation. Video class agnostic segmentation is defined as the task of segmenting objects without regards to its semantics combining appearance, motion and geometry from monocular video sequences. It is crucial first to identify how to formulate this problem. Since the main goal is to segment obstacles without regards to its semantics, we choose two main formulations: (1) The **motion segmentation** formulation that will identify moving objects with no regards to its



Figure 5.1: Video Class Agnostic Segmentation allows to identify objects outside the closed set of known classes.

semantics. Thus it will allow the identification of unknown moving objects such as animals as shown in Figure 5.1. (2) The **open-set segmentation** formulation that will identify pixels belonging to objects outside the closed set of known classes without identifying its exact semantic class. This is shown in Figure 5.1 for the parking and construction scenarios in CARLA, which will identify static unknown objects. The two formulations have their advantages and disadvantages. Motion segmentation has a better capability to perform well across different data distributions where we are able to segment the unknown class (cow) in IDD dataset [171], although our model is trained on Cityscapes-VPS and KITTI with no animal class. The main disadvantage is that it can only detect unknown moving objects, thus static ones like cones, barriers, and construction related equipment can not be segmented. On the other hand, open-set segmentation formulation such as in [186] and our work

for open-set segmentation, relies on learning the statistics of the known classes and a global unknown constant. Both suffer when operating on a different data distribution than the training one. But it has the advantage of segmenting unknown objects whether static or moving.

In the motion segmentation formulation, our work [146][148] is considered the first to motivate the class agnostic segmentation problem and identifying unknown moving objects in the autonomous driving setting. A patent [148] that was created with Valeo Vision Systems was targeted towards creating the first moving object detection network for autonomous driving trained in an end-to-end manner. Concurrent to our work, another learning based motion segmentation for autonomous driving was proposed in [172]. The closest sub-tasks to our work that focus on motion cues are: (1) Unsupervised video Object Segmentation. (2) Motion Segmentation. Unsupervised video object segmentation literature aims at segmenting the appearance and motion salient object in a video sequence. While motion segmentation is mainly focused on segmenting moving objects regardless of how salient they are. Both literature have been described thoroughly in Section 2.3. We further propose a method to improve the open-set segmentation formulation through contrastive learning with semantic and temporal guidance that can improve the discrimination among known and unknown objects.

Our main contributions are:

- Formalizing the video class agnostic segmentation task in autonomous driving, discussing it as two main formulations with insights on them.
- Provide real and synthetic datasets for autonomous driving for that task [146][144].
- The first work for learning moving object detection trained in an end-to-end manner directed towards autonomous driving in a released patent [148].
- A Computationally efficient motion segmentation network that is able to perform close to the state of the art while performing  $4\times$  speedup [141].

- A novel method to learn video class agnostic segmentation using contrastive learning with region-level semantic and temporal guidance is proposed [145].

### 5.1.1 Unknown Objects segmentation

Since this task has relations to open-set segmentation we review in this section the related work in open-set classification/detection/segmentation. Open-set segmentation in autonomous driving has not been thoroughly studied in the literature with only two work in the literature [186] [109]. Wong et. al. [186] inspired from prototypical networks for few-shot learning [153] through learning a prototype per semantic stuff class and thing instance. A multi-frame bird’s eye view representation from LIDAR pointclouds is used as input to their model. Osep et. al. [109] proposed a method that utilizes a video sequence of stereo images with 4D generic proposals for autonomous driving and utilizing parallax to identify temporally consistent objects. However, their method that relies on a category-agnostic object proposal network can ignore unknown objects that are present but not labelled in training examples, and has only been evaluated on 150 images for the open-set task in autonomous driving. It is also a computationally intensive method starting with their reliance on a two-stage object detection method [56], while most of the unknown objects are considered as stuff classes and does not need to identify separate instances/proposals.

Other related open-set detection/classification methods such as [30][185] are either constrained to certain downstream robotic tasks such as robot manipulation or focused on the image classification task, both are much simpler than the segmentation task in autonomous driving scenes. On the contrary, our method focuses on monocular video sequences in autonomous driving scenes. Our open-set segmentation method builds on the model from [186] but focuses on semantic segmentation and learns prototypes per semantic class using both appearance and geometry. We additionally propose contrastive learning with semantic and temporal guidance to improve the unknown objects segmentation which is orthogonal to others contributions and can work with multiple

baselines for open-set segmentation.

### 5.1.2 Contrastive Learning

Since one of our main contributions is on contrastive learning and is mainly inspired by metric learning to benefit the video class agnostic segmentation task we review its related work. Hadsell et. al. [53] was the first to propose a metric learning approach that contrasts positive and negative pairs as the contrastive loss. Dosovitskiy et. al. [36] afterwards developed the pretext task of instance discrimination to learn a representation in a self supervised manner as contrastive learning. There are three aspects to consider in contrastive learning: (1) The type of pretext tasks used in unsupervised methods such as instance discrimination [36][21] or multi-view contrastive coding [163]. (2) Supervised [72] versus Unsupervised [21][36], i.e. whether labels are used to guide the contrastive learning process or not. (3) The contrastive learning mechanism whether it is end-to-end [21], using a memory bank [189], or a momentum encoder as proposed by [55]. Most of the previous literature, contrastive learning was used as a means to leverage unlabelled data and learn in an unsupervised manner. Recently Winkens et. al. [185] proposed a method to improve out-of-distribution(OOD) detection based on a Mahalanobis detector and using embeddings trained in a contrastive learning framework. We extend the method to the task of video class agnostic segmentation in autonomous driving, and propose a temporal version to ensure consistency of the embeddings through representation warping.

## 5.2 Datasets

### 5.2.1 Valeo KITTIMoSeg Dataset

Training convolutional networks requires large amounts of training data. We suggest a pipeline to automatically generate static/moving classification for objects on KITTI dataset [46]. The procedure uses odometry information and annotated 3D bounding boxes for vehicles. The odometry information that includes GPS/IMU readings provides a method to compute the velocity of the

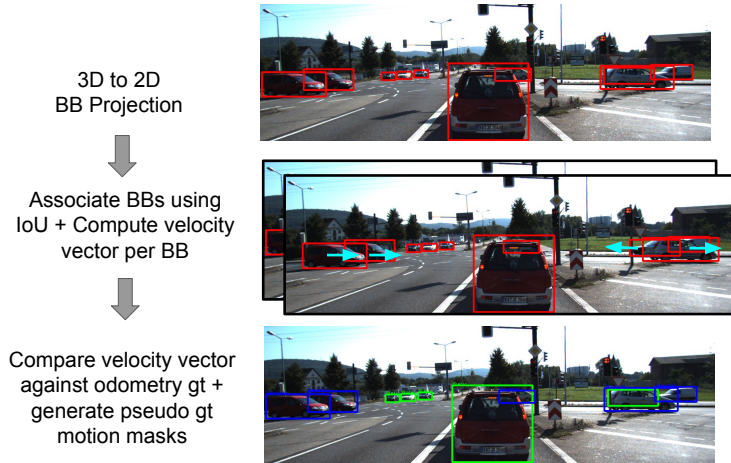


Figure 5.2: Overview of the pipeline used to generate KITTI Moving Object Detection annotations. Blue boxes for moving vehicles, green boxes for static ones.

moving camera. The 3D bounding boxes of the annotated vehicles are projected to 2D images and tagged with their corresponding 3D centroid. The 2D bounding boxes are associated between consecutive frames using intersection over union. The estimated vehicles velocities are then computed based on the associated 3D centroids. The computed velocity vector per bounding box is compared to the odometry ground-truth to determine the static/moving classification of vehicles. The objects that are then consistently identified on multiple frames as moving are kept. In this dataset, the focus is on vehicles with car object category.

An overview of the labeling procedure is shown in Figure 5.2. This is applied on six sequences from KITTI raw data [46] to generate a total of 1750 frames. In addition to these frames, 200 frames from KITTI scene flow are used to provide us with 1950 frames in total. This new dataset was further amended with motion segmentation masks using a trained DeepLab-V3 output masks that overlap with the output moving bounding boxes which we called KITTIMOSeg. For some statistics on the dataset, the total number of static vehicles is 5997, while the number of moving ones is 2383. The dataset is publicly available as part of KITTI Raw Dataset [140] to act as a benchmark on motion detection on KITTI. Although there exists other motion segmentation

Table 5.1: Comparison of different datasets for motion or primary object segmentation. Seqs: Sequences, Cats: Categories, Inst: Instances. Pan: Panoptic, Tr: Tracks.

Dataset	# Frames	# Seqs	# Cats	Inst.	Pan.	Tr.
DAVIS [14]	6208	90	78	✓	✗	✓
KITTI MOTION [172]	455	-	1	✗	✗	✗
KITTI MoSeg [146][122]	12919	~ 38	1	✗	✗	✗
Inst. KITTI MoSeg[100]	12919	~ 38	5	✓	✗	✗
City Motion [172]	3475	-	1	✓	✓	✗
VCAS-Motion (ours)	11008	520	8	✓	✓	✓

datasets such as [116][98][108]. However, they are either synthetic[98], relatively small [108] or have limited camera motion [116] unlike what is present in autonomous driving scenes. Our dataset was further extended to around 13,000 frames from Valeo team by Rashed et. al. work [122] using the same label generation described above but on the full KITTI raw dataset.

## 5.2.2 Wayve Datasets

In collaboration with Wayve we built both real-world and simulation datasets curated for the class agnostic segmentation task. Since we have two main formulations for the problem, we provide real-world dataset for the motion segmentation and synthetic dataset for the open-set segmentation formulation.

### Motion Segmentation Dataset

In the first formulation for the video class agnostic segmentation as a motion segmentation problem, we provide motion annotations extended on the original KITTIMOTS [175] and Cityscapes-VPS [73] datasets. A trajectory annotation tool was built to annotate the trajectories for either moving or static and is further used to provide instance-wise motion masks. Table 5.1 shows the different statistics for our dataset in comparison to other collected datasets in the literature that are used in both motion segmentation and primary object segmentation. Our dataset has the main advantage of having larger variety of object categories and  $50\times$  increase in the video sequences which are the most important aspect for video class agnostic segmentation. In order to push the



Figure 5.3: Curated Wayve motion dataset, extended annotations to real-world data from KITTI and Cityscapes. Red: Moving, Blue: Static.

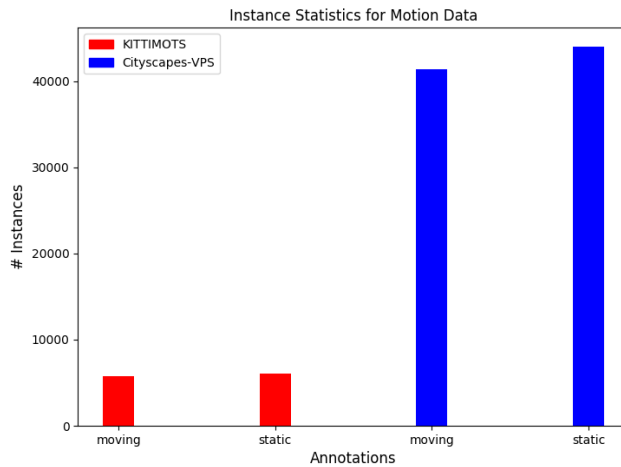


Figure 5.4: Wayve Motion Dataset Statistics.

model to depend more on the motion information and only use the appearance information to detect objectness rather than detecting cars only. Although DAVIS provide a larger variety of object categories the kind of camera motion and scenes in these sequences are much simpler and easier to segment unlike autonomous driving setting. Since autonomous driving scenes mostly have constant fast camera motion, multiple degenerate cases for objects moving parallel to camera motion and cluttered scenes with multiple moving objects.

We provide instance masks unlike most previous literature which is crucial for motion instance segmentation. Concurrent to our work [100] provided instance-wise motion masks, however we provide manually labelled segmen-



Scenario	Unknown Objects
Parking	Cart with Bags Shopping Trolley Garbage Bin
Construction	Traffic Warning Construction Cone
Barrier	Traffic Pole
Training	Barrel Traffic Cone Traffic Barrier Static (others) Dynamic (others)




Figure 5.5: Different scenarios in CARLA Simulation and objects considered as unknown in our synthetic data.

tation masks unlike their weakly annotated ones. Another important aspect for class agnostic segmentation is the panoptic labels that are provided for 3000 frames in our dataset that can aid the identification of unknown objects and will get the class agnostic head to provide redundant signal for a safety-critical approach. Finally, the dataset has extra tracking label annotations which can further aid in tracking moving objects. Figure 5.3 demonstrates the extended annotations for the real-world datasets on both KITTIMOTS and Cityscapes-VPS sequences. Figure 5.4 further shows the datasets statistics showing the number of moving and static instances in both KITTI-MOTS and Cityscapes-VPS.

### Open-set Segmentation Dataset

In the open-set segmentation formulation we care about providing video sequences along with annotations for unknown objects in different autonomous driving scenarios. Thus, we build different scenarios within the CARLA simulation environment [35]. The goal is to incorporate these scenarios as part of the CARLA challenge for autonomous driving to benefit both perception and policy learning researchers. Such scenarios are integrated in the CARLA challenge scenario runner <sup>1</sup>, to evaluate the robustness of autonomous driving

<sup>1</sup>[https://github.com/carla-simulator/scenario\\_runner](https://github.com/carla-simulator/scenario_runner)

	Training
Towns	Town1 Town2 Town3
# Images	57972

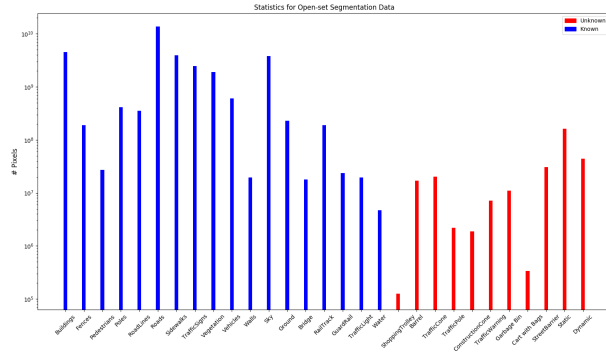


Figure 5.6: Statistics for CARLA for both known classes and unknown objects, the different Towns in CARLA where our dataset was collected for training and testing phase and # Images. Blue: Known Classes, Red: Unknown objects.

systems in terms of safety. Figure 5.5 lists the three main scenarios that are used to evaluate the open-set segmentation task, with objects that are not frequently available in current autonomous driving datasets.

We extend the CARLA environment to provide fine-grained labels for the specific set of unknown objects that are used to analyze the correlation between unknown objects used during training and testing. We further modify the basic driving agent for the ego-vehicle to avoid unknown obstacles, in order to collect large-scale data of up to 50K images with ground-truth depth and semantic segmentation. During collection of training data we use a separate set of unknown objects different than the ones used in testing, and we rather collect in the training scenario with objects randomly placed as obstacles on the road while randomizing both traffic and weather conditions in three different towns. During inference we collect data in the main three scenarios listed in Figure 5.6 in different towns than the ones used in training data collection and with randomized object placement, weather condition and traffic.

## 5.3 Motion Segmentation

### 5.3.1 End-to-End Video Class Agnostic Segmentation

We propose in [146] the first attempt to perform moving object segmentation in automated driving scenes with the focus of using it as a class agnostic segmentation module. This provides an extra means to scale to unknown objects based on their motion representation, unlike large-scale trained semantic segmentation models on a closed set of classes. Our model is shown in Figure 5.7 where we proposed to jointly learn semantic object detection and class agnostic object segmentation relying on appearance and motion information. An encoder-decoder architecture is used for motion segmentation. Similar to the FCN8s [92] architecture, VGG16 network is transformed to a fully convolutional network removing the last fully connected layers. Inspiring from [150][66], a two stream VGG16 is utilized to extract appearance and motion features. The feature maps from both are combined using a summation junction for a memory efficient network. This is followed by two separate decoders one for object detection and another for motion segmentation decoder. The motion segmentation decoder has a  $1 \times 1$  convolutional layer, then three transposed convolutional layers to perform upsampling and a pixel-wise cross entropy loss is used to learn the parameters for that decoder. In order to benefit from high-resolution features, skip connections are used and added to partially upsampled feature maps.

The object detection decoder follows a similar approach to FastBox [162]. It is based on Yolo [127] used as a single shot detector, it has two  $1 \times 1$  convolutional layers. The last layer outputs  $39 \times 12$  grid size representing each cell. The channels in the output layer include the bounding box coordinates, size, and the confidence in the existence of a vehicle. Finally, the rezoom layer is used to overcome the loss of resolution caused by pooling. ROI pooling from the higher resolution layers is followed by  $1 \times 1$  convolutional layers. Then the residuals on the coordinates are regressed over, for a more accurate localization. The loss function used for the detection head combines the L1 loss

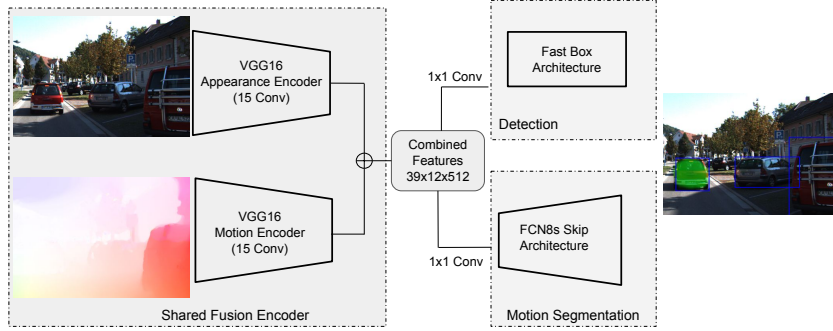


Figure 5.7: MODNet Two Stream Multi-Task Learning Architecture for joint motion segmentation and object detection. Optical Flow and RGB input, RGB image with overlay motion segmentation in green and detected bounding boxes in blue.

for the bounding box regression, with cross entropy for the confidence score. Our model combining appearance and motion features where the appearance stream helps in segmenting the object boundary, while the motion stream identifies moving vehicles. Two different inputs for the motion stream are considered and compared: (1) Optical flow. (2) Image pair of frame  $I_t, I_{t-1}$ . In the latter case, the network is expected to learn an embedding that matches the input image pair. In case of optical flow, middlebury colorwheel is used to convert it to RGB flow[5].

### 5.3.2 Real-time Video Class Agnostic Segmentation

In order to explore design choices for a computationally efficient motion segmentation, a system that decouples the method for feature extraction and decoding method is used for a principled benchmarking [142]. This benchmarking system includes feature extraction architectures VGG-16 [151], ResNet-18 [17], MobileNet [62], and ShuffleNet [202]. Our decoding methods which perform in-the-network upsampling and act as a meta-architecture include SkipNet [92], UNet [130], and Dilation Frontend [199].

**Decoding Meta-Architectures:** The methodology used in the upsampling greatly impacts both accuracy and computational performance. SkipNet architecture denotes a similar architecture to FCN8s [92]. The main idea of the skip architecture is to benefit from feature maps from higher resolution to

improve the output segmentation. This reduces the upsampling factor from  $32\times$  to  $8\times$ . SkipNet applies transposed convolution on heatmaps in the label space instead of performing it on the feature space. This entails a more computationally efficient decoding method than others. U-Net architecture denotes a stage-wise decoding method as it up-samples features using transposed convolution corresponding to each downsampling stage. The up-sampled features are fused with the corresponding features maps from the encoder with the same resolution. The stage-wise upsampling provides higher accuracy than one shot  $8\times$  upsampling of SkipNet. Dilation Frontend architecture utilizes dilated convolution instead of downsampling the feature maps. Dilated convolution enables the network to maintain an adequate receptive field, but without degrading the resolution from pooling or strided convolution. The original dilation frontend [199] removes pooling layers and replaced by dilated convolution in consecutive layers with a dilation factor of 2. However, computational cost increases, since all operations are performed on higher resolution feature maps than the ones from performing pooling.

**Feature Extraction Architectures:** Computational efficiency is the main focus of the work thus only computationally efficient backbones are studied. The first backbone we experiment with is ResNet-18 [17] which incorporates the usage of residual blocks that directs the network towards learning the residual representation over an identity mapping. MobileNet [62] architecture is based on depthwise separable convolution. It is considered an extreme case of the inception module, where separate spatial convolution for each channel is applied. Then  $1 \times 1$  convolution with all the channels to merge the output denoted is applied. The separation in depthwise and pointwise convolution improves the computational efficiency. ShuffleNet encoder [202] is based on grouped convolution that is a generalization of depthwise separable convolution. It uses channel shuffling to ensure the connectivity between input and output channels. This eliminates connectivity restrictions imposed by grouped convolutions.

**Our Proposed Real-time Class Agnostic Model:** The previous comparison of different encoders and decoders is a principled method to motivate

our final architectural design as will be described in the experiments section. Our architecture for motion segmentation is a two-stream ShuffleNet [202] encoding method with SkipNet [92] decoding method for in the network up-sampling. Figure 5.8 demonstrates the detailed architectural design. The modalities fusion denotes the fusion between motion and appearance features. Feature concatenation is used for the modalities fusion, to learn weighted fusion of them. The features fusion denotes the fusion of up-sampled lower resolution features and higher resolution feature maps through element-wise addition. Similar to the previous model we use optical flow from [65] and converted to RGB as in [5].

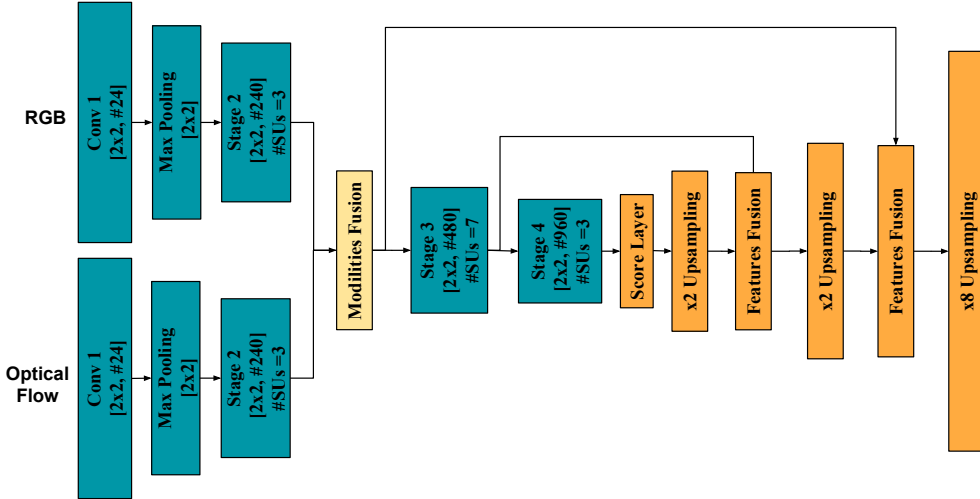


Figure 5.8: Detailed Architecture of Two Stream ShuffleSeg for motion segmentation. SU: ShuffleNet Unit, #: denotes the number of channels. Modalities Fusion: fusion of appearance and motion features through concatenation of features. Feature Fusion: fusion of upsampled lower resolution feature maps and higher resolution maps. Blue: encoding layers, orange: decoding layers, yellow: modalities fusion.

### 5.3.3 Real-time Panoptic and Class Agnostic Segmentation

The previous models do not perform full scene understanding through semantic segmentation nor do they have awareness of the object instances. These two

are crucial in autonomous driving when the model is expected to learn to encode the scene into a meaningful representation and instance-wise masks can enable further tracking and prediction of traffic behaviour. Thus we propose a complete scene understanding method that performs panoptic segmentation i.e. combines instance and semantic segmentation while learning class agnostic embeddings to segment instances outside the closed set of known classes.

### **Backbone with Feature Pyramid Network (FPN)**

We build upon the real-time one-shot instance segmentation model SOLO [179] the model is composed of ResNet-50 [57] backbone, a feature pyramid network [85] and two branches for learning both category and instance masks. Other efficient backbones can replace ResNet-50, a feature pyramid network is used to enable segmentation of different sizes objects where it combines the semantically stronger but low resolution feature maps with the semantically weaker but high resolution feature maps. It takes as inputs  $c_2, c_3, c_4, c_5$  output from the 4 ResNet-50 residual blocks output feature maps. As shown in Figure 5.9 it projects it to 256-D features and upsamples the lower resolution then performs element-wise summation with higher resolution feature maps similar to U-Net but with an added lateral connections to perform prediction on every single resolution level. The output from FPN is bilinearly up/down sampled to form 5 resolution levels with strides 8, 8, 16, 32, 32 which is used as input to the SOLO head.

### **Decoupled SOLO Instance Segmentation Head**

The SOLO head is designed as two branches one for predicting category per element in a square grid with multiple grid sizes  $S \times S$  that are set as hyper-parameters. In our case we use 80, 72, 64, 48, 32 grid sizes. The other branch predicts an instance mask corresponding to every element in the square grid, thus resulting in  $S^2$  mask predictions. A CoordConv [90] layer is used to condition the mask prediction on the corresponding position of the grid to make it spatially variant. The CoordConv layer concatenates on the original features the different positions normalized in the interval  $[-1, 1]$ . Figure 5.10 illustrates

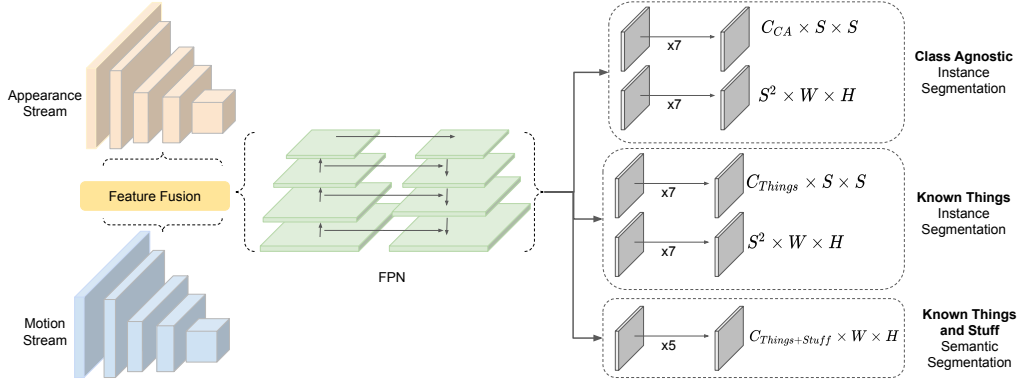


Figure 5.9: Detailed Architecture for Class Agnostic and Panoptic Segmentation in Autonomous Driving.

the SOLO head design where the category branch outputs  $S \times S \times C$  probability maps where  $C$  is the number of classes. While the mask branch predicts  $S^2 \times W \times H$  binary masks where  $W$  and  $H$  the original image size. Thus, the instance segmentation head is fully convolutional and does not depend on box anchors unlike previous methods and is computed in a single shot manner which makes it computationally efficient. Both branches are composed of 7 convolutional modules with RELU and group normalization [187].

For the sake of memory efficiency we use the decoupled SOLO head version which splits the masks prediction for  $S^2$  grid into two branches each with  $S$  masks corresponding to two axes. Since the predicted instance masks are usually sparsely located it is possible to perform this separation. Then the output mask for any element in the grid is the element-wise multiplication of this element in the two xy-branches. Thus, the output from the decoupled version is  $2S \times W \times H$  as shown in Figure 5.10, unlike the vanilla version which outputs  $S^2 \times W \times H$  this makes it memory efficient. We utilize focal loss [86] for the predicted instance category and dice loss [99] for the predicted masks. The labels are assigned using center sampling where a grid element  $(i, j)$  is assigned the instance category if it lies within certain range to the center.

The decoupled SOLO head is responsible for predicting the masks of the things classes (i.e. classes with different instances such as car, pedestrian, bus, ...etc). As for the stuff classes (i.e. classes that do not differentiate



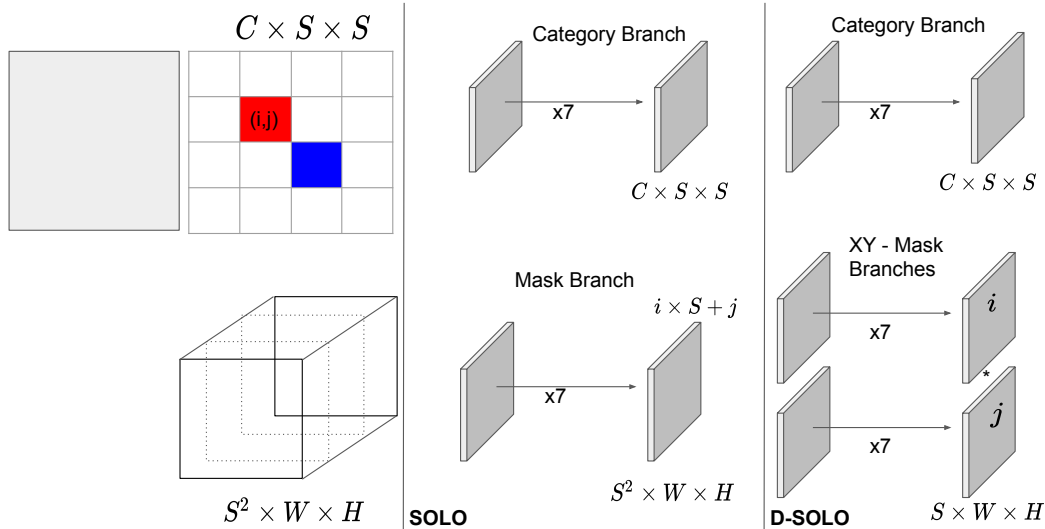


Figure 5.10: Adopted from [179] Vanilla SOLO versus decoupled SOLO heads with detailed explanation for mask prediction branch.

into different instances such as sidewalk, road, ...etc) an extra segmentation head is used to perform pixel-wise classification on the whole scene to capture these. The extra segmentation head takes as input the merged FPN features to the highest resolution level and it has 5 convolutional modules with RELU and group normalization. This is followed by bilinear upsampling and  $1 \times 1$  convolution to predict the probability maps for the full stuff and things classes. A heuristic method is used to perform panoptic fusion between stuff and things masks in order to output the final panoptic masks.

### Class Agnostic Head with Ego-Flow Suppression

We continue to use a two-stream backbone and use input from optical flow computed with FlowNet2 [65]. However, optical flow represents motion attributed to both moving objects and camera motion. Thus we perform ego-flow suppression following [172] through subtracting the computed 2D ego flow from the flow computed from FlowNet2. The ego flow is computed using both depth  $z$  and pose  $R, T$  predicted from a model that is trained in a self supervised manner [50]. The ego-flow is computed following Equation 5.1:

$$x' = K^{-1}x, \tag{5.1a}$$



Figure 5.11: (a) Computed output from ego-flow suppression. (b) Original flow.

$$\tilde{x} = K[R|T]X', \quad (5.1b)$$

$$O_{ego} = \tilde{x} - x, \quad (5.1c)$$

$$O_{supp} = O - O_{ego}, \quad (5.1d)$$

where  $x$  is the 2D homogeneous image coordinates and  $K$  is the camera intrinsics matrix. It projects the 2D homogeneous image coordinates into camera coordinates and applies the transformation computed from the relative pose between two frames then re-projects it back to image coordinates.

The output from ego-flow suppression in comparison to the original computed flow is shown in Figure 5.11, where the moving car’s flow in second row hasnt been suppressed unlike the static parked cars. Since we use a dedicated class agnostic head we are able to train it with separate datasets than the known semantic heads for performing panoptic segmentation. This separation as well allows for ensuring the class agnostic head does not overfit to a certain set of known classes unlike the semantic head. In order to avoid degradation in panoptic segmentation performance we freeze the semantic heads while training the class agnostic head.

## 5.4 Open-Set Segmentation

Our method is shown in Figure 5.12, where the following sections will describe the details.

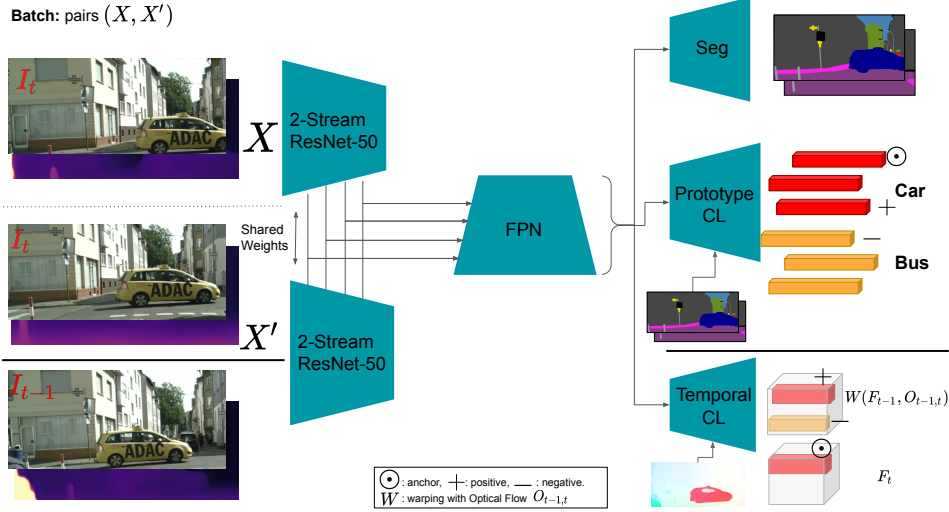


Figure 5.12: Video Class Agnostic Segmentation using Contrastive Learning.

### 5.4.1 Analysis on Unknown Objects

The first question to assess in the open-set segmentation problem is whether the unknown objects that are used during the training phase are correlated with the unknown objects used during testing. Answering this question can provide a better understanding of the task difficulty and the open-set segmentation model scalability. We propose to use the region-level features from masked average pooling following Equation 5.2:

$$P_c = \sum_{i=1}^N \sum_{x,y} \mathbb{1}[M_i^{x,y} = c] F_\theta^{x,y}(X_i), \quad (5.2a)$$

$$d(i, j) = \|P_i - P_j\|_2, \quad (5.2b)$$

where  $F$  is the first two residual blocks from ResNet-50 pretrained on imagenet,  $M$  is the semantic segmentation mask,  $X_i$   $i$ th input image with  $N$  total number of images in the dataset and  $P_c$  denotes the region-level feature for class  $c$ . A small scale data with fine-grained annotations for all unknown objects is collected and region-level features per class are computed then a pair-wise distance measure  $d(i, j)$  is computed among classes. It is used in an agglomerative clustering and a dendrogram among the classes is visualised to

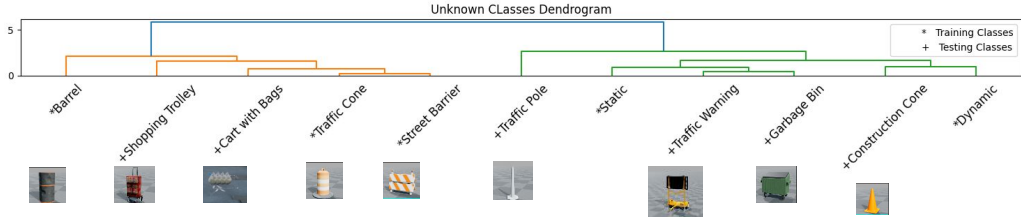


Figure 5.13: Dendrogram among unknown objects used during training and testing phases.

understand the correlation among the unknown objects in the training and testing phases. Figure 5.13 demonstrates the dendrogram among all unknown objects, and shows the classes that are visually similar especially in texture are closer, such as street barrier and traffic cone. From the figure we can see less correlation among some of the different unknown objects used during testing and training, confirming the generalization ability of our model when successful in segmenting these unknown objects during the testing phase.

### 5.4.2 Mahalanobis Based Segmentation

The method we use for the open-set segmentation is similar to [186] but focused only on semantic segmentation without incorporating instances, we learn a representative prototype  $\mu, \sigma$  per class. Then we use a distance similar to the Mahalanobis based distance to classify every pixel based on the matching prototype. We have two baseline models one that relies on appearance only, and another that relies on appearance and geometry. The baseline that fuses the two modalities uses a two-stream backbone model  $f_\theta$  with ResNet-50 backbone [57] and a feature pyramid network [85], which takes as input appearance and depth. This is followed by a semantic segmentation head  $f_\phi$  with 4 convolutional modules with ReLU and group normalization, that learns prototypes  $\mu, \sigma$  per class.

Let the input appearance and depth be denoted as  $x$ , the extracted features  $h = f_\theta(x)$ , and the embeddings from the segmentation head  $m = f_\phi(h)$ . The semantic segmentation head predicts the class probabilities following Equa-

tion 5.3:

$$d_{i,k} = \frac{-\|m_i - \mu_k\|^2}{2\sigma_k^2}, \quad (5.3a)$$

$$d_{i,C+1} = \gamma, \quad (5.3b)$$

$$\hat{y}_{i,k} = \frac{\exp(d_{i,k})}{\sum_{j=1}^{C+1} \exp(d_{i,j})}, \quad (5.3c)$$

$$l_{seg} = \frac{-1}{N} \sum_{i=1}^N \sum_{k=1}^{C+1} y_{i,k} \log \hat{y}_{i,k}, \quad (5.3d)$$

where the distance  $d_{i,k}$  denotes the distance of pixel  $i$  features to the representative prototype of class  $k$ . A global learnable constant  $\gamma$  is used to estimate the unknown objects regardless of the objects' semantics, then a softmax over  $C + 1$  distances is used to estimate the probability of the pixel to belong to a certain class.

### 5.4.3 Contrastive Learning with Semantic and Temporal Guidance

In this section we detail our contrastive learning method that is shown in Figure 5.12. Our models have an input batch  $\{x, y\}_{i=1}^N$ , on which we apply random augmentations to get  $\{x', y'\}_{i=1}^N$ . The random augmentations itself differs based on semantic or temporal guidance being used as detailed in their respective sections. Following the literature [72] we call this the multiviewed batch  $\{x, y\}_{i=1}^N \cup \{x', y'\}_{i=1}^N$  and is used as input to our model to end up with  $2N$  images per batch. An extra contrastive learning head  $f_\alpha$  is used to improve the separation of known classes and unknown objects, similar to the segmentation head we use 4 convolutional modules with ReLU and group normalization. Both the segmentation and contrastive learning head share the same backbone encoder and feature pyramid network. The embeddings that are computed from the contrastive learning head are denoted as  $z = f_\alpha(h)$ .

In order to improve the segmentation of known classes and more importantly the unknown objects we propose to perform contrastive learning on the

prototype-level. We use semantic guidance to define a region as the segmentation mask of a certain class within the image, and extract prototypes using masked average pooling. The contrastive learning with semantic guidance follows the Equations:

$$\beta_c = \sum_{i=1}^N \sum_{x,y} \mathbb{1}[M_i^{x,y} = c] z^{x,y}, \quad (5.4a)$$

$$L_{pcl} = \sum_{i=1}^B \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\beta_i \cdot \beta_p / \tau)}{\sum_{a \in A(i)} \exp(\beta_i \cdot \beta_a / \tau)}. \quad (5.4b)$$

where  $\beta_c$  is the output from masked average pooling on feature embeddings  $z$  to represent class  $c$  region. This results in a set of pairs  $\{\beta_i, y_i\}_{i=1}^B$  with  $y_i$  the corresponding class of the prototype  $\beta_i$ . This set is either sampled from a memory queue with the previous batches features or from the current batch. In Equation 5.4  $P(i)$  is the indices of the positive regions as  $P(i) = \{p \in A(i) : y_p = y_i\}$  and  $A(i)$  is the set of indices of all prototypes. The final segmentation loss follows Equation 5.5.

$$L = L_{seg} + \lambda L_{pcl}, \quad (5.5)$$

We propose another variant that employs temporal relations among embeddings from a video sequence in an unsupervised manner without the need of semantic segmentation labels. It is based on the concept of representation warping with optical flow to ensure temporal consistency of embeddings [45]. Previous work in contrastive learning worked mainly on the video-level and was not concerned with the pixel-wise embeddings. However, since our main downstream task is segmentation we rather use representation warping with the estimated optical flow between two consecutive frames. Then compute the contrastive loss on the aligned regions from both frames belonging to a video sequence. It follows the Equations:

$$F'_{t-1} = W(F_{t-1}, O_{t-1,t}) \quad (5.6a)$$

$$\delta_t = \text{AP}(F_t), \delta'_{t-1} = \text{AP}(F'_{t-1}) \quad (5.6b)$$

$$L_{tcl} = \sum_{x,y} -\log \frac{\exp(\delta_t(x,y) \cdot \delta'_{t-1}(x,y)/\tau)}{\sum_{x',y'} \exp(\delta_t(x,y) \cdot \delta'_{t-1}(x',y')/\tau)} \quad (5.6c)$$

where AP is an average pooling operation, and  $F_t, F_{t-1}$  is the features extracted for the current and previous frame. The final loss is:

$$L = L_{seg} + \lambda L_{tcl}, \quad (5.7)$$

## 5.5 Experimental Results

### 5.5.1 Experimental Setup

Throughout experiments on pixel-wise class agnostic segmentation, the Adam optimizer [74] is used with learning rate  $1 \times 10^{-5}$  for KITTI-MoSeg experiments,  $1 \times 10^{-4}$  for real-time benchmarking experiments. L2 regularization is used in the loss function, with  $5 \times 10^{-4}$  factor. The encoder is initialized with Imagenet pretraining weights for all experiments. Weighted cross entropy loss from [113] is used in the real-time benchmarking experiments, to overcome the class imbalance. The class weight is computed as  $w_{class} = \frac{1}{\ln(c+p_{class})}$ . In the real-time benchmarking experiments a width multiplier of 1 for MobileNet is used to include all the feature channels. The number of groups used in ShuffleNet is three, as previous results [202] recommended.

For the later work on panoptic and class agnostic segmentation, and for the open-set segmentation work we follow this setup. Our models are implemented using PyTorch library [114]. Throughout all experiments we use SGD with momentum optimizer with 0.005 learning rate and 0.9 momentum, and weight decay of  $1 \times 10^{-3}$  with 15 epochs. A step learning rate scheduling which reduces the learning rate with 0.1 at epochs 6 and 8. We use random augmentations as random scales  $\{0.8, 1.3\}$ , random flipping and random cropping with  $320 \times 512$  as crop sizes. In the open-set segmentation loss we use  $\lambda = 0.5$  and merge the feature pyramid network output through upsampling to the largest resolution and averaging the different scales.

### 5.5.2 Datasets

Along with our collected KittiMoSeg and Wayve datasets that are described earlier in Section 5.2. We use additional three datasets to help benchmark with other methods or to gain extra variability in object categories.

**Cityscapes [25]:** We use Cityscapes dataset [25] in order to perform proper benchmarking for different semantic segmentation models in a unified benchmark. The dataset contains 5000 images with fine annotation, with 20 classes including the ignored class. Another section of the dataset contains coarse annotations with 20,000 labeled images. These are used in the case of Coarse pre-training that improves the results of the segmentation.

**City-Kitti-Motion [172]:** In our real-time motion segmentation experiments when comparing to SMSNet [172] we train on City-Motion with 3475 frames and test on KITTI-Motion with 200 frames, we augment City-Motion with extra annotations from KITTI-MoSeg that do not overlap with the sequences for 200 frames.

**DAVIS’17 [14]:** In order to provide higher variability in object categories that we test against we use the video object segmentation benchmark from DAVIS2017 which includes object instances annotations. The dataset include 90 sequences for 78 object categories including different animals that can help ensure the generalization ability of our class agnostic segmentation head.

**IDD [171]:** is collected on Indian roads and has 10,004 images finely annotated with 34 classes from 182 drive sequences. We specifically use classes that are outside the set of classes in CityscapesVPS for qualitative evaluation.

**Cityscapes-VPS [73]:** this is the video sequences version of cityscapes, where for every 30 frame sequence that corresponds to one image in cityscapes validation set, 6 frames are densely labelled for panoptic segmentation. We use this dataset for both panoptic segmentation evaluation and to conduct experiments for the open-set segmentation formulation by taking out some classes during training and using them as testing. The overall dataset has



2400 training images and 300 validation ones that we use as a hold out test set.

### 5.5.3 Evaluation Metrics

The evaluation metrics used in class agnostic and semantic segmentation are precision, recall, F-score and mean intersection over union (IoU). The evaluation metric used for detection in pixel-wise class agnostic segmentation experiments is mean average precision(mAP) and average precision (AP) for static/moving classes. In the pixel-wise class agnostic segmentation average precision of car class is also measured showing different difficulties for easy, medium, and hard setup as in KITTI benchmark [47]. Note that it is important to evaluate the static/moving classification standalone without including errors from the detection itself. The average precision used is computed on the detected bounding boxes that match bounding boxes from the ground truth.

As for the panoptic and class agnostic segmentation experiments since they take into account different instances we rather use segmentation quality, recognition quality and panoptic quality metric as reported in [75] for the panoptic segmentation. We also evaluate similar to the panoptic the same three metrics including class agnostic quality metric (CAQ) following the work from [186]. On the other hand, the open-set segmentation we evaluate using mean IoU and class agnostic IoU. Finally, for evaluating the computational performance we report inference time in milliseconds and floating point operations (GFLOPS).

### 5.5.4 Results and Discussion

#### Class Agnostic Segmentation Results on KITTI-MoSeg

Table 5.2 shows a comparison of the different models that show the benefit from the use of optical flow as an explicit motion representation instead of learning it implicitly using a Siamese model with image pair input. In these experiments we used the initial small version of KITTI-MoSeg with 1300 frames only and the joint model that performs pixel-wise motion segmentation and vehicle detection. The two-stream (RGB+OF) shows a 10% increase in mean average

Table 5.2: Quantitative evaluation on KITTI MoSeg data for our proposed joint detection and motion segmentation network.

	AP Static	AP Moving	mAP
MODNet (image pair)	<b>60.7</b>	44.29	52.5
MODNet (RGB+OF)	58.6	<b>66.54</b>	<b>62.6</b>

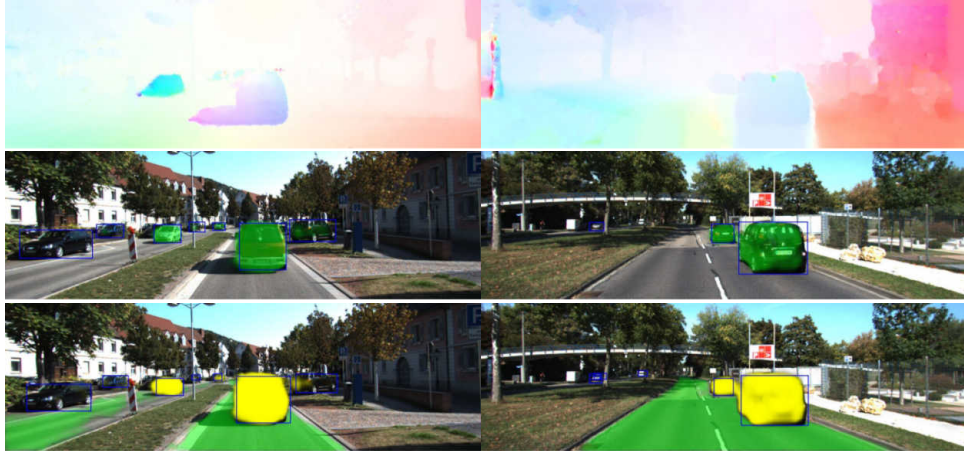


Figure 5.14: Qualitative evaluation on KITTIMoSeg data for our proposed two-stream multi-task learning network MODNet. top row: Input Optical Flow, middle row output of 2 tasks: overlay motion mask (green), bottom row output of 3 tasks: overlay motion mask (yellow), road segmentation (green) and detected bounding boxes (blue).

precision over the one using image pair as input. The model taking image-pair input struggles more than (RGB + OF), since optical flow readily computes the motion explicitly and has been trained on large-scale data regardless of semantics. Figure 5.14 shows the qualitative results for our proposed MODNet that was jointly trained for vehicle detection, motion segmentation and road segmentation.

### Real-time Class Agnostic Segmentation Benchmarking Results

In this section an ablation study using the decoupled design is applied to observe the accuracy-speed trade-off for different design choices. This aids the selection of the two-stream architectural design further used in our method. Table 5.3 shows the results for the ablation study on Cityscapes validation set with different encoders-decoders reporting mIoU, GFLOPs to demonstrate the trade-off between accuracy and computations. The UNet method of in-

Table 5.3: Comparison of different encoders and decoders on **Cityscapes validation set**. Evaluation is in terms of intersection over union and GFLOPs on image resolution  $1024 \times 512$ . Coarse indicates whether the network was pre-trained on the coarse annotation or not.

Encoder	Decoder	Coarse	GFLOPs	mIoU
SkipNet	MobileNet	No	13.8	61.3
SkipNet	ShuffleNet	No	<b>4.63</b>	55.5
UNet	ResNet18	No	43.9	57.9
UNet	MobileNet	No	55.9	61.0
UNet	ShuffleNet	No	17.9	57.0
Dilation	MobileNet	No	150	57.8
Dilation	ShuffleNet	No	71.6	53.9
SkipNet	MobileNet	Yes	13.8	<b>62.4</b>
SkipNet	ShuffleNet	Yes	<b>4.63</b>	59.3

crementally upsampling with-in the network provides the best in terms of mIoU. However, SkipNet architecture is more computationally efficient with  $4\times$  reduction in GFLOPs. This is explained by the fact that transposed convolutions in UNet are applied in the feature space rather than the label space as in SkipNet, which has lower dimension. It also shows the benefit from pretraining the models with coarse annotations first then finetuning on the smaller set with fine annotations on cityscapes. Dilation frontend following the design from [199] is adapted to the different backbones, it shows increase in computational cost as the operations are performed on larger resolution feature maps.

Experimental results on the cityscapes test set are shown in Table 5.4. ENet [113] is compared to SkipNet-ShuffleNet and SkipNet-MobileNet in terms of accuracy and computational cost. Both SkipNet-ShuffleNet and SkipNet-MobileNet outperform SegNet [2] in terms of computational cost and accuracy with reduction up to  $143\times$  in GFLOPs. This motivated the final two-stream architectural design that was based on ShuffleNet encoder and SkipNet meta-architecture.

Experiments on KITTI-Motion and our KITTI-MoSeg are conducted to evaluate both accuracy and computational performance. Table 5.5 shows comparison to the state of the art on KITTI-Motion, our method outperforms

Table 5.4: Comparison to the state of the art segmentation networks on **Cityscapes test set** in terms of intersection over union, and GFLOPs on image resolution **640x360**.

Model	GFLOPs	IoU	iIoU
SegNet [3]	286.03	56.1	34.2
ENet [113]	3.83	58.3	24.4
SkipNet-VGG16 [92]	445.9	<b>65.3</b>	<b>41.7</b>
SkipNet-ShuffleNet (ours)	<b>2.0</b>	58.3	32.4
SkipNet-MobileNet (ours)	6.2	61.5	35.2

Table 5.5: Quantitative results on KITTI-Motion in terms of mean intersection over union, mean average precision, running time and frame-rate on image resolution  $384 \times 768$ .

Model	mIoU	mAP	Time	fps
GEO-M[79]	48.15	-	-	-
AHCRF+Motion [126]	68.0	-	-	-
CRF-M [126]	77.9	-	240,000	0.004
SmSNet [172]	<b>84.1</b>	<b>86.4</b>	<b>105</b>	9
ShuffleSeg (RGB+OF) (ours)	68.8	78.0	<b>27</b>	<b>37</b>

SmSNet [172] in terms of running time with  $4\times$  speedup to reach 36 ms instead of 153 ms on image resolution  $384 \times 768$ . This speedup enables it to run the motion segmentation part real-time on Nvidia Jetson TX2 for embedded vision with 9 fps. Our network still performs with comparable accuracy in comparison to SmSNet [172], as shown in Figure 5.15 for precision and recall curve, yet with a  $4\times$  speedup in frame rate. That facilitates its deployment for autonomous driving and driving assisted systems where real-time can be crucial for actuation control or alerting the driver to moving objects.

### Real-time Class Agnostic and Panoptic Segmentation Results

Our final model is trained with freezing the panoptic segmentation head, the feature pyramid network and backbone and training the class agnostic head solely. The reason for this choice is not to lead to the degradation of semantic tasks. We leave for future work experiments on learning the weighting between losses for both panoptic and class agnostic heads. Table 5.6 shows the results of our proposed multi-task panoptic and class agnostic segmentation which

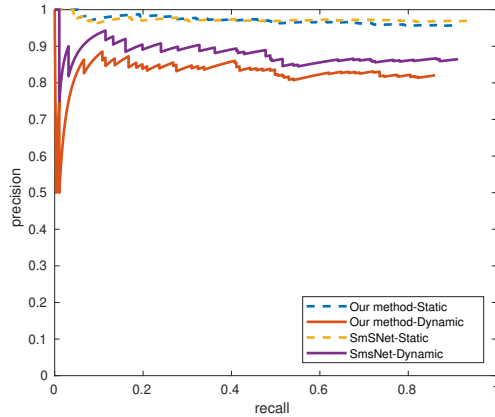


Figure 5.15: Precision-Recall Curve on KITTI-Motion. Our method provides on par results in detection (PR) with  $4\times$  speedup.

we term as VCAS-V1, the Pan-Base denotes our panoptic baseline without performing class agnostic segmentation. We initially report the class agnostic baseline trained on DAVIS [14] and compare it with the VCAS multi-task model to show that it maintains comparable class agnostic quality V1 stands for the version trained for motion segmentation. Our VCAS model with ego-flow suppression improves further the CAQ metric on KITTIMOTS motion data. Our results act as a baseline for different approaches that tackle class agnostic instance segmentation, and shows how motion and geometric cues can leverage the class agnostic segmentation further on.

Figure 5.16 shows qualitative results on KITTIMOTS Motion, Cityscapes VPS Motion and IDD datasets respectively for our class agnostic segmentation head. IDD results show that our VCAS-V1 is able to segment objects outside the closed set of semantic classes, which the panoptic segmentation was trained on, through relying on motion and geometric cues. While our model is still able to maintain the capability to perform panoptic segmentation even cross datasets as trained on CityscapesVPS and evaluated on KITTI and IDD.

### Contrastive Learning with Semantic and Temporal Guidance

In Table 5.7 we evaluate our baseline segmentation network trained with all classes on Cityscapes, to evaluate the baseline performance on the general segmentation task without unknown objects. Then we start with conducting

Table 5.6: Results of CA and panoptic segmentation model. Tr-Te: Training and Test data used for CA segmentation. EFS: Ego Flow Suppression. D: DAVIS, K: KITTIMOTS Motion, C: Cityscapes-VPS Motion. Time is measured on image resolution  $1024 \times 2048$  in seconds.

Model	Tr-Te	EFS	CA Metrics			Panoptic Metrics		
			SQ	RQ	CAQ	$PQ_{All}$	$PQ_{Th}$	$PQ_{St}$
Pan-Base	-	-	-	-	-	56.4	49.9	61.1
CA-Base	D-D	✗	76.6	60.6	46.4	-	-	-
VCAS-V1	D-D	✗	72.2	53.2	38.6	56.4	49.9	61.1
VCAS-V1	KC-K	✗	82.4	72.2	59.5	56.4	49.9	61.1
VCAS-V1	KC-K	✓	<b>82.4</b>	<b>75.4</b>	<b>62.1</b>	56.4	49.9	61.1
VCAS-V1	KC-C	✗	77.9	57.3	44.7	56.4	49.9	61.1
VCA-V1	KC-C	✓	78.0	56.2	43.8	56.4	49.9	61.1

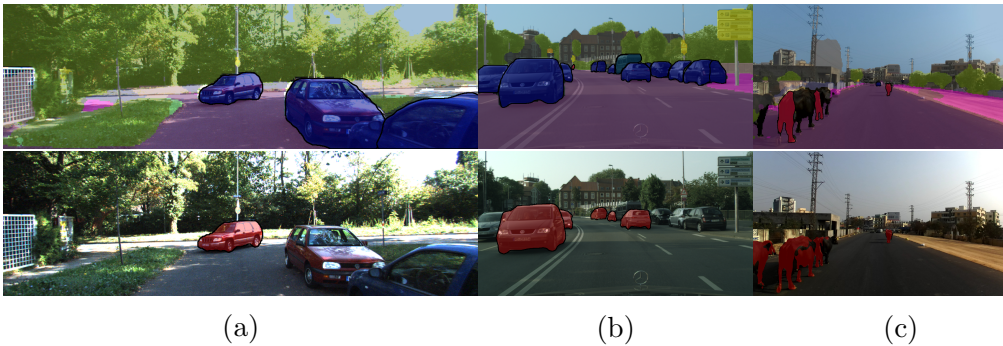


Figure 5.16: Top: predicted panoptic segmentation. Bottom: predicted CA segmentation on (a) KITTIMOTS Motion. (b) Cityscapes-VPS Motion. (c) IDD.

experiments on Cityscapes-VPS, and compare four main variants: (1) No contrastive learning (CL) baseline. (2) Image level (CL) after performing global average pooling on the features. (3) Prototype level with semantic guidance. (4) Aligned regions level with temporal guidance. Table 5.7 shows the results for two sets of experiments with batch size 4 and 2 on Cityscapes-VPS. Due to limited GPU memory we were not able to conduct experiments for the temporal guidance variant using the larger batch size. However, the effective batch size used in the contrastive loss is different. So a batch size of 2 will result in an effective batch size of 35 in the prototype-level variant, which is the prototypes extracted in an image on average along with the ones from the memory queue. As for the temporal variant it will result in 100, which is the

Table 5.7: Open-Set Segmentation (VCAS-V2) Results on Cityscapes-VPS and CARLA. Fully supervised: Training the segmentation head on all cityscapes classes without a learnable global constant for the unknown object. CA-IoU: class agnostic IoU on the unknown objects.

Method	Dataset	Batch	mIoU	CA-IoU
Fully Supervised	Cityscapes [25]	4	<b>65.5</b>	-
No CL	Cityscapes-VPS [73]	4	63.2	17.9
Prototype CL			<b>63.7</b>	<b>18.7</b>
No CL	Cityscapes-VPS [73]	2	56.4	18.4
No CL + SimCLR[21] Pre			56.1	17.3
Image CL			60.1	19.8
Prototype CL			<b>62.7</b>	<b>21.5</b>
Temporal CL			61.7	21.4
No CL	CARLA	2	<b>45.7</b>	<b>41.9</b>
Prototype CL			44.2	37.2

dimensions of the feature map output from average pooling.

In both sets of experiments the prototype-level CL improves the CA-IoU. The temporal CL improves over the baseline, but does not outperform the prototype-level variant. However, the temporal variant has the advantage that it does not need any semantic guidance unlike the prototype-level variant. Our experiments show as well that both semantic and temporal guidance improves over the use of image-level CL. Since our main task is video segmentation it is not sufficient to contrast embeddings globally. In summary, our initial experiments show that the segmentation task for both known and unknown classes benefits from the auxiliary contrastive loss especially with semantic guidance. Since semantic guidance improves the discrimination between known and unknown classes, while temporal guidance ensures the temporal consistency of the embeddings.

Finally, we pick the best CL variant which is performed on the prototype-level and compare to the baseline on our collected CARLA dataset. Initial results, show unlike Cityscapes-VPS the baseline is not improved with the contrastive loss. There are multiple differences between both datasets regarding the size where CARLA is  $25\times$  larger than Cityscapes-VPS and has more pixels labelled as unknown. In an upcoming experiments we show that a re-

Scenario	CA-IoU
Barrier	22.1
Construction	41.4
Parking	27.0

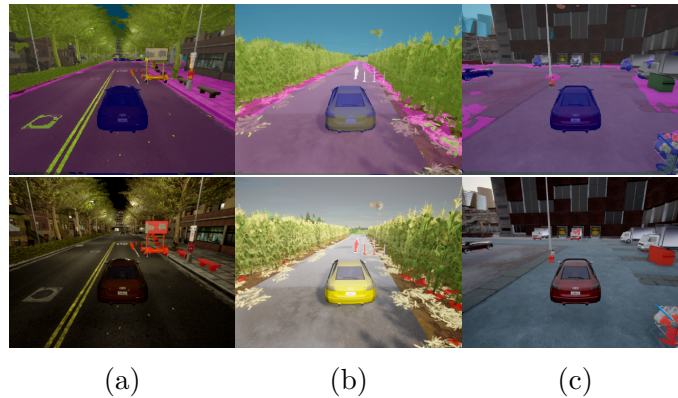


Figure 5.17: CA-IoU reported per scenario. Predicted semantic and class agnostic segmentation on CARLA Scenarios (a) Construction. (b) Barrier. (c) Parking. Top: semantic segmentation. Bottom: class agnostic segmentation.

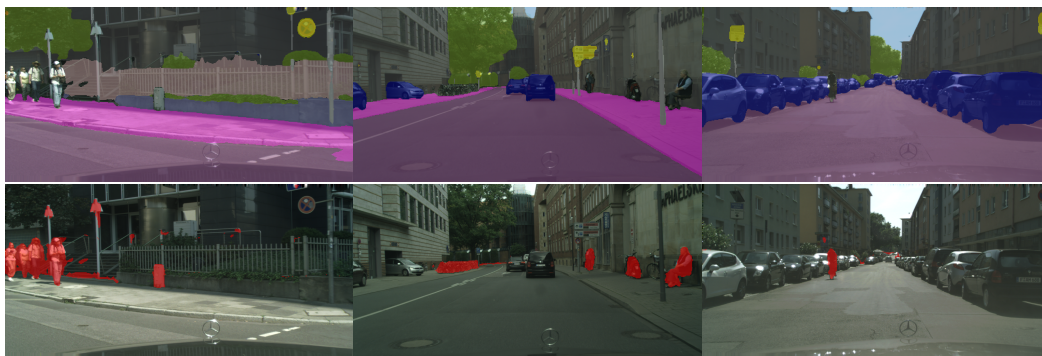


Figure 5.18: Predicted semantic and class agnostic segmentation on Cityscapes-VPS. Top: semantic segmentation. Bottom: class agnostic segmentation (Note: pedestrian, rider, bicycle and motorcycle are withheld from training).

duced set of CARLA will lead to the same conclusions as Cityscapes-VPS experiments. Figure 5.18 shows the results for segmenting both known classes and unknown objects on Cityscapes-VPS. It demonstrates the model ability to segment bicycle and motorcycle that was not previously seen during training. Figure 5.17 shows the results on the CARLA scenarios which confirms on the model’s ability to segment some of the unknown objects that did not appear during training such as in the Parking Scenario. Some of these objects Garbage Bin and Traffic Warning have less relation to unknown objects used during the training phase as shown in Figure 5.13.

**What is the effect of the contrastive training on the shared em-**



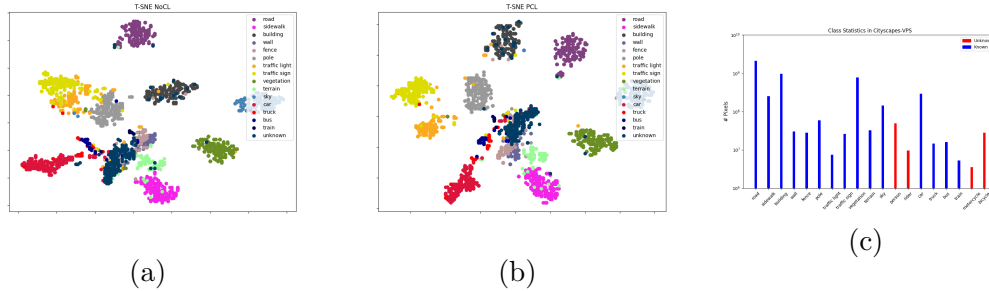


Figure 5.19: T-SNE visualisation of the masked embeddings for 15 known classes along with the unknown objects. (a) No Contrastive Learning. (b) Prototype-level Contrastive Learning. (c) Class statistics in Cityscapes-VPS.

**beddings:** Figure 5.19 shows the T-SNE [96] visualisations for both the baseline without contrastive learning versus the contrastive learning with semantic guidance variant. The embeddings from the feature pyramid network are masked with the groundtruth masks of the different semantic classes to extract prototypes and gone through dimensionality reduction for visualisation. It shows how the embeddings from the contrastive learning on the prototype-level (semantic guidance) are better clustered and separated especially in severe class imbalance cases. The Figure shows the statistics per class to demonstrate which classes suffer from that. In case of the traffic sign, traffic light, pole classes the baseline will lead to confusion among these three classes unlike the contrastive learning variant with better separation. It also leads to better separation of the unknown objects from the known classes such as class “Terrian”.

**Do we need the auxiliary loss during training?** In Table 5.7 we show the results for the baseline (No CL) but rather using pretrained weights from SimCLR [21] versus the different contrastive learning variants. It confirms on the need to have the auxiliary loss during training of the segmentation head to improve the discrimination between known and unknown objects. Table 5.8 ablates the factor  $\lambda$  by which we balance the main segmentation loss and the auxiliary contrastive loss. It shows generally smaller factor is better, as a factor of 1.0 degrades the segmentation of known classes. We rather use 0.2 throughout all the experiments.

Table 5.8: Auxiliary loss factor for the Prototype and Temporal CL.

	$\lambda$	mIoU	CA-IoU
Prototype CL	0.2	<b>62.8</b>	20.9
	0.5	62.7	21.5
	1.0	57.5	<b>22.5</b>
Temporal CL	0.2	<b>61.6</b>	<b>19.9</b>
	0.5	56.9	17.6

Table 5.9: Comparison between different Average Pooling Factors in Temporal CL.

AP Factor	mIoU	CA-IoU
0.05	61.6	19.9
0.1	<b>61.7</b>	<b>21.4</b>
0.3	62.2	20.5

**The effect of a memory queue in contrastive learning with semantic guidance:** Table 5.10 clearly shows the need for including a memory queue during contrastive training to increase the effective batch size for the contrastive loss. Since our encoder is a two stream model that combines depth and appearance it will not be possible to use a momentum encoder due to practical limitations in the GPU memory and computational resources required. Thus, using a memory queue with limited capacity is best to use in our case to ensure that only features from the latest iterations are preserved.

**The effect of average pooling in the temporal contrastive learning:** Table 5.9 demonstrates the segmentation accuracy for both known classes and unknown objects with different average pooling factors. The average pooling factor is the factor multiplied by the original feature map size and used as the kernel size for the pooling. The smaller the factor the higher the resolution of the output. The results show that smaller factors are better as it will lead to contrasting smaller regions temporally, this can correspond to different object parts. Unlike a larger factor of 0.3 that leads to the loss in resolution and confusion among features that can correspond to different semantic objects.

**Reduced Variability in Objects labelled as Unknown during Training:** In the main experiments we initially use polynomial learning rate scheduling, since it is commonly used in state of the art methods [20]. However, we

Table 5.10: Effect of Memory Queue in Prototype CL.

	mIoU	CA-IoU
No Memory	59.3	<b>21.3</b>
Memory Queue	<b>62.8</b>	20.9

Table 5.11: Quantitative Results on Cityscapes-VPS with larger batch size (4) and Step Learning Rate Scheduling. FD: Full Data. LU: Less number of objects labelled as unknown during training.

Data Mode	Method	Batch	mIoU	CA-IoU
FD	No CL	4	<b>63.1</b>	21.0
	PCL	2	62.7	<b>21.5</b>
	PCL	4	63.0	21.3
LU	No CL	4	63.1	18.8
	PCL	2	<b>63.4</b>	<b>20.0</b>

further validated that using step learning rate scheduling improves the baseline with a significant margin. We show in Table 5.11 the results for experiments using step learning rate scheduling and larger batch size (4). It shows that contrastive learning with semantic guidance improves the CA-IoU but with a minimal gain of 0.5%. However, more experiments on reducing the number of objects labelled as unknown demonstrate the benefit from using contrastive learning with semantic guidance on CA-IoU. In these sets of experiments only pixels belonging to class “Person” or originally ignored in Cityscapes are labelled as unknown during training.

### Why Prototype and Temporal Contrastive Learning Improves?

The main reasons behind prototype and temporal contrastive learning improvement are two fold. In the prototype contrastive learning as detailed in [72] the semantic guidance will lead to increased positives and negatives. This consequently leads to improvement in the discrimination between signal (i.e. the prototype of a certain class) and noise (i.e. negative prototypes of others including unknown objects). In [72] it was shown that the contrastive loss, whether supervised or unsupervised, learns to perform hard negative mining implicitly. It does so by increasing the gradient contribution of hard examples, and decreasing it for the easy examples. It was also shown that specifically for supervised contrastive learning, increasing the positives and negatives leads

to an increase in the gradients contribution when dealing with hard positives. These two main reasons explain the improvement from prototype-level contrastive learning over the baseline, especially in case of lower variability in the unknown objects used to train the global constant.

As for the temporal contrastive learning, the improvement stems from constraining the features to be temporally consistent. Since the contrastive loss samples one region as an anchor and its aligned region (i.e. warped representation using optical flow) as positive, while the remaining regions are sampled as negatives. The contrastive loss with increased negatives will, as discussed previously, increase the gradients contribution from hard positives. Thus, it will perform better in aligning the features than simply merging features from both frames as input to the segmentation head.

**Investigating the Discrepancy between Cityscapes-VPS and Synthetic data Results:** In the synthetic dataset (CARLA) we found that the baseline without contrastive learning outperforms the contrastive learning with semantic guidance. So we use a reduced version of the synthetic dataset with 2400 images similar to Cityscapes-VPS dataset size. Then we reduce the pixels labelled as unknown during training to only two objects. Table 5.12 demonstrates that, less data and less number of objects labelled as unknown during training, leads to clear gain from the contrastive learning with semantic guidance. This conforms with the above experiments on Cityscapes-VPS as well. In summary, our proposed auxiliary contrastive loss is more suitable and will lead to improvements when facing problems with relatively medium-scale data for the known classes and less variability in the objects labelled as unknown during training. It leaves an open question on how the auxiliary contrastive loss can improve even with abundant data. For our future work, we want to explore how to segment unknown objects without learning the global constant that was also used in [186].

Table 5.12: Quantitative Results on CARLA with less data and less number of objects labelled as unknown. LD: Less Data (2400 frames similar to Cityscapes-VPS). LU: number of pixels labelled as unknown during training.

Data Mode	Method	mIoU	CA-IoU
LD + LU	NoCL	38.5	6.5
	PCL	<b>41.7</b>	<b>16.0</b>

## 5.6 Summary

In this chapter we formalized the video class agnostic segmentation task and provided necessary datasets and benchmarks for the two main tracks: (1) motion segmentation track, and (2) open-set segmentation track. In the motion segmentation track, the first motion segmentation dataset in autonomous driving is proposed, and an improved version for instance segmentation is provided. Multitask models that combine semantic and class agnostic segmentation/detection are benchmarked and publicly released. For the open-set segmentation track our synthetic dataset provides the means to assess model scalability to unknown objects not previously seen during training and further study the relation among unknown objects used between training and testing. Finally, a novel contrastive learning with semantic and temporal guidance that suits the dense prediction in video sequences is proposed and has shown to outperform the baseline model.

# Chapter 6

## Motion Adaptation for Video Object Segmentation

### 6.1 Introduction

Continuing on the intuition of using motion cues as a way to segment unknown objects, we further propose to use these predictions as pseudo-labels. We propose a motion adaptation method that is based on computing a distance transform on the computed probability maps from class agnostic segmentation. These are further used to adapt a model responsible for segmenting this novel class instance through different video sequences. Our method goes through three phases: (1) **Base Training Phase**: in which a teacher model is trained to perform motion segmentation in a class agnostic manner using large-scale video object segmentation data. (2) **Motion adaptation Phase**: in which the teacher model’s output probability maps are used to compute pseudo labels using a distance transform to the highly confident positive pixels. Then it is further used to adapt a student model. (3) **Inference Phase**: in which the student model is used to segment that specific object instance through different sequences. Figure 6.1 shows an overview of the proposed method. The two main reasons behind using the pseudo-labels from the teacher model is: (1) The student model is more computationally efficient. The inference and adaptation time for the teacher model is 1.5x of the student model’s. The adaptation occurs only once on the first frame, then the more efficient student model can be used for inference. (2) The teacher model can be used to

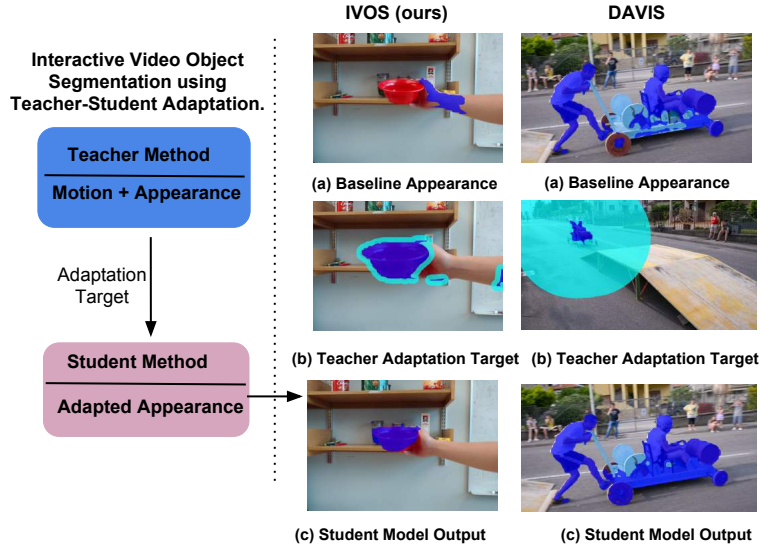


Figure 6.1: Overview of the proposed Teacher-Student adaptation method for video object segmentation. The teacher model based on motion cues is able to provide pseudo-labels to adapt the student model. Blue: confident positive pixels. Cyan: ignored region in the adaptation.

generate pseudo-labels for the potential object of interest. It does not require the human to provide manual segmentation mask during the teaching phase which provides a natural interface to the human. On the other hand, if motion is characteristic of the class, then it is useful cue for subsequent identification from the adapted student model. If the adapted model was still dependant on optical flow it will only be able to recognize the object in motion.

Video semantic segmentation for robotics is widely used in different applications such as autonomous driving [25][131], and robot manipulation [32][70]. Object segmentation can aid in grasping, manipulating objects, and learning object affordances [32]. In robot manipulation, learning to segment new objects incrementally, has significant importance. Real world environments have far more objects and more appearance variation than can be feasibly trained a-priori. Current large-scale datasets such as Image-Net [31] do not cover this. A recent trend in robotics is toward human-centered artificial intelligence. Human-centered AI involves learning by instruction using a human teacher. Such human-robot interaction (HRI) mimics children being taught novel concepts from few examples [97]. In the robotic setting, a human teacher

demonstrates an object by moving it and showing different poses, while verbally or textually teaching its label. The robot is then required to segment the objects in other settings where it is either static or manipulated by the human or the robot itself. An interesting extension is for the robot to experiment with the object without human intervention (e.g., roll it, pick-up, etc.) to observe its characteristic behaviour. We demonstrated the aforementioned HRI setting in our team submission to the KUKA Innovation Challenge at the Hannover Fair [129]. This HRI setting has few differences to conventional video object segmentation: (1) Abundance of the different poses of the object. (2) The existence of different instances/classes within the same category. (3) Different challenges introduced by cluttered backgrounds, different rigid and non-rigid transformations, occlusions and illumination changes. In this chapter, we focus on these robotics challenges and provide a new dataset and a new method to study such a scenario.

We collected a new dataset to benchmark (I)nteractive (V)ideo (O)bject (S)egmentation in the HRI scenario, which would act as a good testing platform for the idea of motion adaptation mentioned earlier. The dataset contains two types of videos: (1) A human teacher showing different household objects in varying poses for interactive learning. (2) Videos of the same objects used in a kitchen setting while serving and eating food. The objects occur both as static objects and active objects being manipulated. Manipulation was performed by both humans and robots. The aim of this dataset is to facilitate incremental learning and immediate use in a collaborative human-robot environments, such as assistive robot manipulation. Datasets that have a similar setting such as ICUBWorld transformations dataset [112], and the Core50 dataset [91] were proposed. These datasets include different instances within the same category. They benchmark solutions to object recognition in a similar HRI setting but do not provide segmentation annotations unlike our dataset. Other datasets were concerned with the activities of daily living such as the ADL dataset [118]. The dataset was comprised of ego-centric videos for activities. However, such ADL datasets do not contain the required teaching videos to match the HRI setting we are focusing on. Table 6.1 summarizes the most



Table 6.1: Comparison of different datasets. T:Turntable, H:handheld.

Dataset	Sess.	Cat.	Obj.	Acq.	Tasks	Seg.
RGB-D [134]	-	51	300	T	✗	✗
BIG BIRD [152]	-	-	100	T	✗	✗
ICUB 28 [111]	4	7	28	H	✗	✗
ICUB World [112]	6	20	200	H	✗	✗
Core50 [91]	11	10	50	H	✗	✗
<b>IVOS</b>	12	12	36	H	✓	✓

relevant datasets suited to the HRI setting.

The main contribution of our collected IVOS dataset is providing the manipulation tasks setting with objects being manipulated by humans or a robot. In addition to providing segmentation annotation for both teaching videos and manipulation tasks. It enables researchers to analyze the effect of different transformations such as translation, scale, and rotation on learning video object segmentation. It acts as a benchmark for interactive video object segmentation in the HRI setting.

Our proposed method inspires from teacher-student training but rather employs it for cross-modal fine-tuning on the pseudolabel object masks. A recent survey of different teacher-student training methods was provided by Gou et al. [52] which included our work on the video object segmentation application. Our method outperforms the state-of-the-art on the popular DAVIS’16 [115] and FBMS [108] benchmarks with 6.8% and 1.2% in F-measure respectively. On our new IVOS dataset results show the motion adapted network outperforms the baseline with 46.1% and 25.9% in mIoU on Scale/Rotation and Manipulation Tasks respectively. Our code <sup>1</sup> and IVOS dataset <sup>2</sup> are publicly available. A video description and demonstration is available <sup>3</sup>. Our main contributions are :

- Providing a Dataset for Interactive Video Object Segmentation (IVOS) in a Human-Robot Interaction setting, and including manipulation tasks unlike previous datasets.

<sup>1</sup>[https://github.com/MSiam/motion\\_adaptation](https://github.com/MSiam/motion_adaptation)

<sup>2</sup><https://msiam.github.io/ivos/>

<sup>3</sup><https://youtu.be/36hMbAs8e0c>

- A teacher-student adaptation method is proposed to learn new objects from a human teacher without providing manual segmentation labels. We propose a novel pseudo-label adaptation based on a teacher model that is dependant on motion. Adaptation with discrete and continuous pseudo-labels are evaluated to demonstrate different adaptation methods.

## 6.2 (I)nteractive (V)ideo (O)bject (S)egmentation (IVOS) Dataset

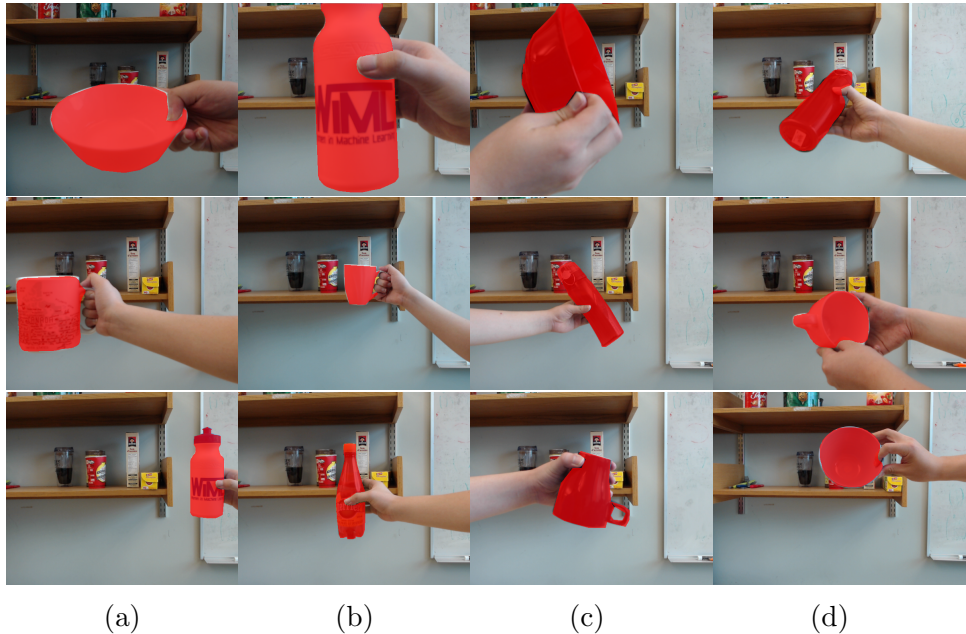


Figure 6.2: Samples of collected Dataset IVOS, Teaching Objects Setting. (a) Translation split. (b) Scale split. (c) Planar Rotation split. (d) Out-of-plane Rotation.

We collected IVOS for the purpose of benchmarking (I)nteractive (V)ideo (O)bject (S)egmentation in the HRI setting. We collect the dataset in two different settings: (1) Human teaching objects. (2) Manipulation tasks setting. Unlike previous datasets in human robot interaction IVOS dataset provides video sequences for manipulation tasks. In addition to providing segmentation annotation for both teaching videos and manipulation tasks.

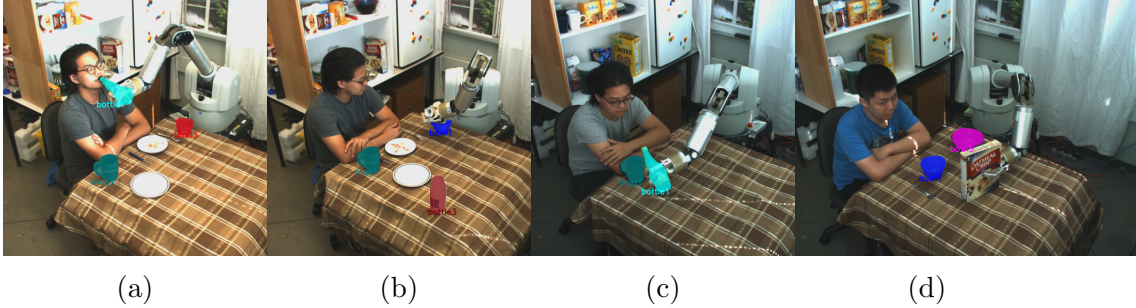


Figure 6.3: Samples of collected IVOS dataset, Robot manipulation Tasks Setting with segmentation annotation. Manipulation Tasks: (a) Drinking. (b) Stirring. (c) Pouring Water. (d) Pouring Cereal.

### 6.2.1 Human Teaching Objects

For teaching, videos are collected while a human moves an object with her hand. The unstructured human hand motion naturally provides different views of the object and samples different geometric transformations. We provide transformations such as translation, scale, planar rotation, out-of-plane rotation, and other transformations such as opening the lid of a bottle. Two illumination conditions are provided: day-light and indoor lighting, which sums up to 10 sessions of recording for both illumination and transformations. Figure 6.2 shows a sample for the objects being captured under different transformations with the segmentation masks. In each session a video for the object held by a human with relatively cluttered scene background is recorded.

A GRAS-20S4C-C fire-wire camera is used to record the data along with a Kinect sensor [156]. The collected data is annotated manually with polygonal masks using the VGG Image Annotator tool [39]. The final teaching videos contain 12 object categories, with a total of 36 instances under these categories. The detection crops are provided for all the frames, while the segmentation masks are provided for 20 instances with  $\sim 18,000$  annotated masks.

### 6.2.2 Manipulation Tasks Setting

The manipulation task benchmark includes two video categories: one with human manipulation, and the other with robot manipulation. Activities of Daily Living (ADL) such as food preparation are the focus for the recorded

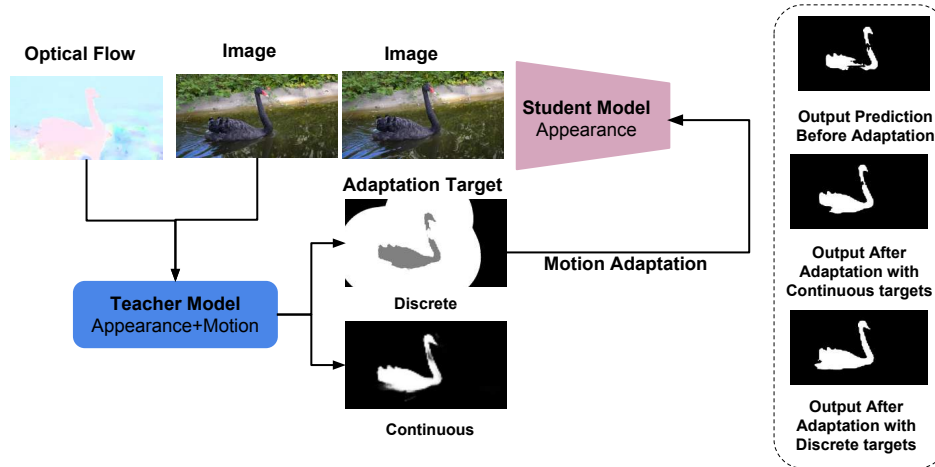


Figure 6.4: Motion Adaptation of fully convolutional residual networks pipeline.

tasks. The aim of this benchmark is to further improve perception systems in robotics for assisted living. Robot trajectories are created through kinesthetic teaching, and the robot pose way-points are provided in the dataset. In order to create typical robot velocity and acceleration, profiles trajectories were generated from these way-points using splines as is standard in robotics.

The collected sequences are further annotated with segmentation masks similar to the teaching objects setting. Figure 6.3 shows some of the recorded frames with ground-truth annotations. It covers 4 main manipulation tasks: *cutting, pouring, stirring, and drinking* for both robot and human manipulation covering a total of 56 tasks. The dataset contains  $\sim 8,900$  frames with segmentation masks, along with the recorded robot trajectories to enable further research on how to learn these trajectories from visual cues.

## 6.3 Motion Adaptation

### 6.3.1 Baseline Network Architecture

The student model in this work is built on the wide ResNet architecture presented in [191]. The network is comprised of 16 residual blocks. Dilated convolution [199] is used to increase the receptive field without decreasing the resolution. The output from the network is bilinearly upsampled to the initial

image resolution. The loss function used is bootstrapped cross entropy [190], which helps with class imbalance. It computes the cross entropy loss from a fraction of the hardest pixels. Pre-trained weights on PASCAL dataset for objectness is used from [176], to help the network generalize to different objects in the scene. Then it is trained on DAVIS training set, the student model without adaptation is denoted as the baseline model throughout the paper.

The teacher network incorporates motion from optical flow, where a two-stream wide ResNet for motion and appearance is used. Each stream contains 11 residual blocks for memory efficiency reasons. The output feature maps are combined by multiplying the output activation maps from both motion and appearance streams. After combining features another 5 residual blocks are used with dilated convolution. The input to the motion stream is the optical flow computed using [88], and converted into RGB representation using HSV encoding [5].

### 6.3.2 Motion Adaptation using Pseudo-labels

There is an analogy between this work and the work in [169], where a student method is learning to mimic a teacher method. In our work the teacher method is a motion dependent one, and the student method tries to mimic the teacher during inference through motion adaptation. The teacher-student training helps the network understands the primary object in the scene in an unsupervised manner. Unlike the work in [176] that first fine-tunes the network based on the manual segmentation mask then adapts it on-line with the most confident pixels. Our method provides a natural human robot interaction that does not require manual labelling for initialization.

Our approach provides two different adaptation methods, adapting based on discrete or continuous labels. The teacher network pseudo-labels are initially filtered to remove parts representing the human moving using the output human segmentation from Mask R-CNN [56]. When discrete labels are used it is based on pseudo-labels from the confident pixels in the teacher network output. Such a method provides superior accuracy, but on the expense of tuning the parameters that determine these confident pixels. Another method

that utilizes continuous labels adaptation from the teacher network is also introduced. This method alleviates the need for any hyper-parameter tuning but on the cost of degraded accuracy. Figure 6.4 summarizes the adaptation scheme, and shows the output pseudo-labels, the output segmentation before and after adaptation.

In the case of discrete pseudo-labels, the output probability maps from the teacher network is further processed in a similar fashion to the semi-supervised method [176]. Initially the confident positive pixels are labeled, then a geometric distance transform is computed to label the most confident negative pixels as shown in Algorithm 1.

---

**Algorithm 1** Motion Adaptation Algorithm.

**Input:**  $X$ : images used for teaching.  $N$ : number of samples used.  $M_{teacher}$ : Teacher Model.  $M_{student}$ : Student Model.

**Output:**  $\hat{M}_{student}$ : Adapted Student Model.

---

```

1: function TEACH( $N, X, M_{teacher}, M_{student}$ )
2:   for  $i$  in  $N$  do
3:      $P_i = M_{teacher}(X_i)$ 
4:      $\hat{M}_{student} = \text{Adapt}(P_i, M_{student})$ 
5:   end for
6: end function

```

**Discrete Labels Adaptation Method**

```

7: function ADAPT( $A_t, M_{student}$ )
8:   Mask  $\leftarrow$  IGNORED
9:   pos_indices  $\leftarrow$  ( $A_t > \text{POS\_TH}$ )
10:  dt  $\leftarrow$  DISTANCE_TRANSFORM(Mask)
11:  neg_indices  $\leftarrow$  ( $dt > \text{NEG\_DT\_TH}$ )
12:  Mask[pos_indices]  $\leftarrow$  1
13:  Mask[neg_indices]  $\leftarrow$  0
14:  return finetune( $M_{student}, \text{Mask}$ )
15: end function

```

---

In the case of continuous labels, the output probability maps are used without further processing. This has the advantage of not using any hyper-parameters or discrete label segmentation. The cross entropy loss can be viewed as a mean to decrease the divergence between the true distribution  $p$  and the predicted one  $q$ . In our case the true distribution is the probability maps from the teacher network, while the predicted is the student network

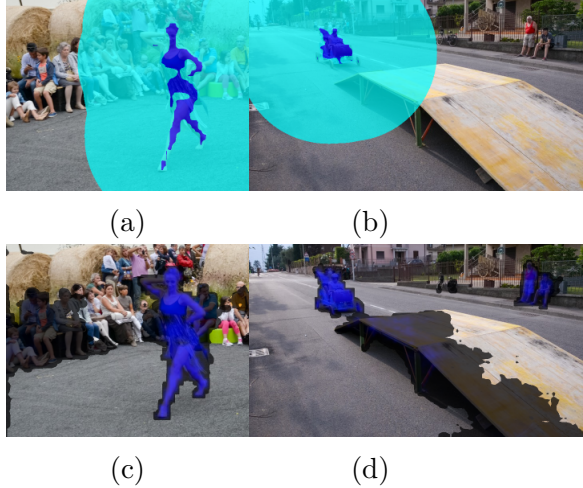


Figure 6.5: (a,b) Discrete adaptation targets (pseudo-labels), cyan is the unknown region, blue is the confident positive pixels. (c, d) Continuous adaptation targets.

output. Figure 6.5 shows the difference between the pseudo-labels for both discrete and continuous variants. Conditional random fields is used as a post-processing step on DAVIS’16 and FBMS.

## 6.4 Experimental Results

### 6.4.1 Experimental Setup

For all experiments the DAVIS’16 training data is used to train our Appearance model and the Appearance+Motion model. The optimization method used is Adam [74] with learning rate  $10^{-6}$  during training, and  $10^{-5}$  during on-line adaptation. In on-line adaptation 15 iterations are used in the scale/rotation experiments and 50 in the tasks experiments. Adaptation is only conducted once at the initialization of the video object segmentation. The positive threshold used to identify highly confident positive samples is 0.8, and the negative threshold distance to the foreground mask is 220 in case of DAVIS’16 benchmark. Since IVOS is recorded in an indoor setup, a negative distance threshold of 20 is used.

Table 6.2: Quantitative comparison on DAVIS’16 benchmark. MotAdapt-1: Continuous Labels, MotAdapt-2: Discrete Labels.

Method	$\mathcal{J}$			$\mathcal{F}$		
	Mean	Recall	Decay	Mean	Recall	Decay
NLC[41]	55.1	55.8	12.6	52.3	51.9	11.4
SFL[23]	67.4	81.4	6.2	66.7	77.1	5.1
LMP [165]	70.0	85.0	1.3	65.9	79.2	2.5
FSeg [66]	70.7	83.5	1.5	65.3	73.8	1.8
LVO [165]	75.9	89.1	<b>0.0</b>	72.1	83.4	1.3
ARP [77]	76.2	<b>91.1</b>	<b>0.0</b>	70.6	83.5	7.9
Baseline (ours)	74.0	85.7	7.0	74.4	81.6	<b>0.0</b>
MOTAdapt-1 (ours)	75.3	87.1	5.0	75.3	83.8	3.3
MOTAdapt-2 (ours)	<b>77.2</b>	87.8	5.0	<b>77.4</b>	<b>84.4</b>	3.3

Table 6.3: Quantitative results on FBMS dataset (test set).

Method	$\mathcal{P}$	$\mathcal{R}$	$\mathcal{F}$
FST [110]	76.3	63.3	69.2
CVOS [161]	83.4	67.9	74.9
CUT [71]	83.1	71.5	76.8
MPNet-V[165]	81.4	73.9	77.5
LVO[165]	<b>92.1</b>	67.4	77.8
Base (ours)	80.8	76.1	78.4
ours (ours)	80.7	<b>77.4</b>	<b>79.0</b>



Figure 6.6: Qualitative Evaluation on the FBMS dataset. Top: LVO [165]. Bottom: ours.

## 6.4.2 Generic Video Object Segmentation

In order to evaluate the performance of our proposed motion adaptation (MotAdapt) method with respect to the state-of-the-art, we experiment on generic video object segmentation datasets. Table 6.2 shows quantitative analysis on DAVIS’16 benchmark compared to the state-of-the-art unsupervised methods.



Table 6.4: mIoU on IVOS over the different transformations and tasks. IVOS dataset teaching is conducted on few samples from the translation, then evaluating on scale, rotation and manipulation tasks. MotAdapt-1: Continuous Labels. MotAdapt-2: Discrete Labels.

Model	Scale	Rotation	Manipulation Tasks
Baseline	14.5	13.8	14.7
MotAdapt-1	63.8	49.5	30.2
Mot-Adapt-2	<b>69.0</b>	<b>51.5</b>	<b>40.6</b>

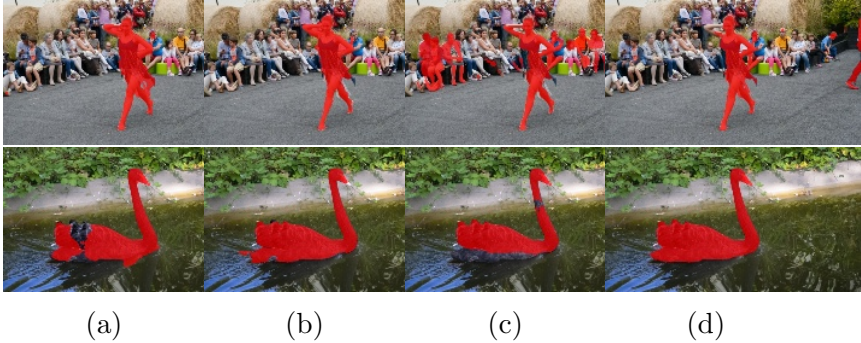


Figure 6.7: Qualitative evaluation on DAVIS'16. (a) LVO [165]. (b) ARP [77]. (c) Baseline. (d) MotAdapt.

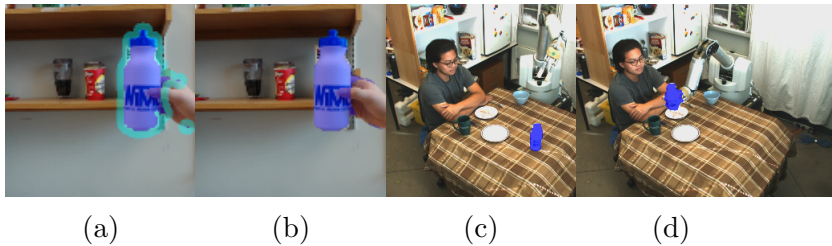


Figure 6.8: Qualitative evaluation on IVOS Manipulation Tasks Setting. (a) Teaching Phase, Discrete Labels. (b) Teaching Phase, Continuous Labels. (c) Inference Phase before manipulation. (d) Inference Phase, during manipulation.

One of the variants of MotAdapt based on discrete labels outperforms the state of the art with 6.8% in F-measure, and 1% in mIoU. Table 6.3 shows quantitative results on FBMS dataset, where our MotAdapt outperforms the state of the art with 1.2% in F-measure and 10% in recall.

Figure 6.6 shows qualitative results on FBMS highlighting the improvement gain from motion adaptation compared to LVO [165]. Figure 6.7 shows qualitative evaluation on DAVIS'16, where it demonstrates the benefit from

motion adaptation compared to the baseline (top row), and compared to LVO [165] and ARP [77] (bottom row).

### 6.4.3 Video Object Segmentation in HRI Setting

Our method is evaluated in the HRI scenario on our dataset IVOS. The teaching is performed on the translation sequences, with only the first two frames used to generate pseudo-labels for adaptation. An initial evaluation is conducted on both scale and rotation sequences, in order to assess the adaptation capability to generalize to different poses and transformations. Table 6.4 shows the comparison between the baseline method without adaptation, and the two variants of motion adaptation on the scale, rotation and tasks sequences. The discrete and continuous variants for our motion adaptation outperform the baseline with 54.5% and 49.3% respectively on the scale sequences. Similarly on the rotation sequences it outperforms the baseline with 37.7% and 35.7% respectively. The main reason for this large gap, is that general segmentation methods will segment all objects in the scene as foreground, while our teaching method adaptively learns the object of interest that was demonstrated by the human teacher.

All manipulation tasks sequences where the category bottle existed is evaluated and cropped to include the working area. Our method outperforms the baseline on the tasks with 25.9%. The first variant of our adaptation method generally outperforms the second variant with continuous labels adaptation. However the second variant has the advantage that it can work on any setting such as DAVIS and IVOS without tuning any hyper-parameters. Figure 6.8 shows the output from our adaptation method when it is recognized by the robot, and while the robot has successfully manipulated that object.

## 6.5 Summary

In this chapter we proposed a novel approach for visual learning by instruction. Our proposed motion adaptation (MotAdapt) method provides a natural interface to teaching robots to segment novel object instances. This enables

robots to manipulate and grasp these objects. Two variants of the adaptation scheme is experimented with. Our results show that Mot-Adapt outperforms the state of the art on DAVIS'16 and FBMS benchmarks, while outperforms the baseline on our collected IVOS dataset.

# Chapter 7

## Conclusion and Future Work

In this chapter we summarize our conclusions and the future directions that can be tackled continuing on what we started and employing the intersection among few-shot and video class agnostic segmentation. Throughout this thesis we have thoroughly investigated the different challenges and solutions shared among few-shot and video object segmentation and formalized the video class agnostic segmentation to benefit from both. We have further driven forward the few-shot object segmentation methods with metric learning and attention mechanisms.

### 7.1 Summary of Contributions

As a summary of the thesis contributions:

- **Advancing Few-shot Object Segmentation Methods:** We have tackled two main issues in few-shot object segmentation literature and provided solutions with experimental results that provide empirical support. The first issue is requiring two sets of parameters in two-branch methods [136]. We instead proposed a single branch method with shared weights used for both support and query sets. Our method that inspired from the connection between softmax classification and proxy NCA proved to be of a significant improvement over the SOA at the time on Pascal-5i. The second issue we addressed in the literature is that they mostly relied on a single vector representation for the support set which

does not capture details necessary for the segmentation problem. We instead proposed a co-attention with semantic conditioning to leverage the interaction of the support and query with only image-level labels. Our method has shown to be competitive with some of the methods that use pixel-level labelled masks, while outperforming other weakly supervised methods including the ones that use bounding box annotation. The semantic conditioning has shown to alleviate ambiguities arising from tendency to segment base classes or other common objects between the support and query sets. We further proposed a setup for temporal few-shot learning and show different ways to investigate the overlap between video object segmentation and few-shot object segmentation.

- **Video Class Agnostic Segmentation:** The most important contribution of this thesis is formalizing the task of video class agnostic segmentation in autonomous driving where the goal is to segment instances of unknown objects through utilizing appearance, motion and geometry. we presented two formulations based on motion segmentation and open-set segmentation, unlike previous work [109][186] that only focused on the open-set segmentation formulation. In the motion segmentation formulation, our model MODNet was the first end-to-end deep learning based method in autonomous driving, and a patent about the work was presented. Additionally, our method for automatic generation of motion annotations on KITTI resulted in KITTI-MoSeg that was followed by multiple works in the field of autonomous driving using it [100] and extending it [122]. In the open-set segmentation, our contrastive learning with semantic and temporal guidance has also offered another way of improving baseline models that can segment unknown objects even if they are static. Our proposed methods although focused on autonomous driving as an application, can be extended to other robotics applications such as robot manipulation.
- **Advancing Unsupervised Video Object Segmentation Methods:** Finally we have presented a method that inspires from semi-supervised

video object segmentation methods through fine-tuning the model on pseudo-labelled masks for the primary object in a video sequence. Our method was inspired by [176] where we proposed a motion adaptation mechanism using a two-stream teacher model to generate the pseudo-labelled masks with a distance transform to compute the confident negative pixels that are further apart from confident positive pixels. Our overall performance was able to set a new record on DAVIS benchmark for unsupervised video object segmentation at the time.

## 7.2 Future Work

For our future work we leave multiple research questions that we think are of significance and hasn't been thoroughly tackled in the literature yet:

- **Temporal Object Segmentation for Few-shot Learning:** Although we have proposed the initial setup there is still plenty of work that can be done in this area that would benefit from assumptions such as pixels that move together belong to the same object, or enforcing temporal consistency of the masked embeddings. Our current setup samples one query image from the video sequence, but a better setup would sample multiple consecutive query images to show the effect of employing such assumptions and constraints to improve few-shot segmentation accuracy. There has been work on transductive few-shot learning that benefits from unlabelled data but it never tackled the temporal constraints that can be used for the unlabelled data in our setting.
- **Generalized Few-shot Object Segmentation:** There has not been fruitful effort in this setup at the time of this thesis work. One of the main directions to solve the issue in which all the base classes are much reduced is through getting the adaptation parameter to be learned through an adaptation generator that is meta-trained with the segmentation model. There has been recent work in that direction that hasn't been published yet and still under review that inspires from our direc-

tion as well on imprinting and adaptation [164]. Generally speaking, few-shot segmentation methods still lack significantly in comparison to fully supervised methods such as DeepLabV3 [20]. Where DeepLabV3 has 85.7% mean intersection over union (mIoU), while the best few-shot method we demonstrate in this thesis is still around 52.1% mIoU on Pascal dataset. This clearly indicates that there is still a longer path ahead for few-shot methods to perform on-par to fully supervised ones, and investigating more on metric learning (contrastive learning especially) and attention mechanisms seems to be a good direction to bridge that gap.

- **Contrastive Learning with Temporal Guidance:** Through our initial efforts in contrastive learning with semantic guidance we see improved performance for segmenting unknown objects. However, when inspecting the temporal guidance it does improve over the baseline with no contrastive learning but does not improve over the semantic guidance one. Although temporal guidance will lead to temporally consistent embeddings that is necessary in any video segmentation task. Thus, one of the main future directions for this work is better interaction between both semantic and temporal guidance to improve the video class agnostic segmentation. Generally, more attention needs to be given to the problem of video class agnostic segmentation in the autonomous driving literature. As currently it is very neglected in the community with few work addressing this issue beside ours [186][109]. That was another motivation for us to build the Carla scenarios to show explicit reasons for why this problem is important instead of vastly labelling all possible classes in a closed set fashion. Although there is tendency in the autonomous driving literature to focus on active learning which will address part of this problem and is definitely an interesting partial solution. But the main question of whether in real-time autonomous driving systems will be able to detect failures in detection/segmentation or unknown objects on the fly is still a vital question in terms of safety.

- **Interpretability of Few-Shot and Video Object Segmentation:**  
All the literature in both tasks lack interpretability that can lead the way to designing better models, and is important for safety critical application such as autonomous driving.

## 7.3 Closing Remarks

There has been limited attention given to the intersection of few-shot and video object segmentation methods in the literature, where we show that multiple shared assumption and solutions can further advance both. Specifically in autonomous driving towards a safety critical approach, it is of utmost importance to have redundancy in the output signal signifying obstacles. This motivated the video class agnostic segmentation task which employs the relation among few-shot and video object segmentation, and for which we encourage more research in that direction.



# References

- [1] A. Antoniou, A. Storkey, and H. Edwards, “Data augmentation generative adversarial networks,” *arXiv preprint arXiv:1711.04340*, 2017.
- [2] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *arXiv preprint arXiv:1511.00561*, 2015.
- [3] —, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [4] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [5] S. Baker, D. Scharstein, J. Lewis, S. Roth, M. J. Black, and R. Szeliski, “A database and evaluation methodology for optical flow,” *International Journal of Computer Vision*, vol. 92, no. 1, pp. 1–31, 2011.
- [6] R. Benenson, S. Popov, and V. Ferrari, “Large-scale interactive object segmentation with human annotators,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 700–11 709.
- [7] L. Bertinetto, J. F. Henriques, P. H. Torr, and A. Vedaldi, “Meta-learning with differentiable closed-form solvers,” *arXiv preprint arXiv:1805.08136*, 2018.
- [8] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, “Visual object tracking using adaptive correlation filters,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010*, IEEE, 2010, pp. 2544–2550.
- [9] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [10] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, “Segmentation and recognition using structure from motion point clouds,” in *European Conference on Computer Vision*, Springer, 2008, pp. 44–57.

- [11] T. Brox and J. Malik, “Object segmentation by long term analysis of point trajectories,” in *European Conference on Computer Vision*, Springer, 2010, pp. 282–295.
- [12] J. S. Bruner, R. R. Olver, P. M. Greenfield, *et al.*, “Studies in cognitive growth.,” 1966.
- [13] S. Caelles, K.-K. Maninis, J. Pont-Tuset, L. Leal-Taixé, D. Cremers, and L. Van Gool, “One-shot video object segmentation,” *arXiv preprint arXiv:1611.05198*, 2016.
- [14] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool, “The 2019 davis challenge on vos: Unsupervised multi-object segmentation,” *arXiv preprint arXiv:1905.00737*, 2019.
- [15] M. Caron, P. Bojanowski, A. Joulin, and M. Douze, “Deep clustering for unsupervised learning of visual features,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 132–149.
- [16] J. Carreira and A. Zisserman, “Quo vadis, action recognition? a new model and the kinetics dataset,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 4724–4733.
- [17] A. Chaurasia and E. Culurciello, “Linknet: Exploiting encoder representations for efficient semantic segmentation,” *arXiv preprint arXiv:1707.03718*, 2017.
- [18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *arXiv preprint arXiv:1606.00915*, 2016.
- [19] ———, “Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834–848, 2017.
- [20] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” *arXiv preprint arXiv:1706.05587*, 2017.
- [21] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” *arXiv preprint arXiv:2002.05709*, 2020.
- [22] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” *CoRR*, vol. abs/1904.04232, 2019. arXiv: 1904.04232. [Online]. Available: <http://arxiv.org/abs/1904.04232>.
- [23] J. Cheng, Y.-H. Tsai, S. Wang, and M.-H. Yang, “Segflow: Joint learning for video object segmentation and optical flow,” *arXiv preprint arXiv:1709.06750*, 2017.

- [24] C. Christensen, J. Upatnieks, and B. Guenther, “Us army missile research and development command,” Technical Report T-79-18, Tech. Rep., 1979.
- [25] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.
- [26] D. Cremers, “Statistical shape knowledge in variational image segmentation,” 2002.
- [27] D. Cremers, M. Rousson, and R. Deriche, “A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape,” *International journal of computer vision*, vol. 72, no. 2, pp. 195–215, 2007.
- [28] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, “Class-balanced loss based on effective number of samples,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9268–9277.
- [29] *Decolonizing artificial intelligence*, <http://blog.shakirm.com/2018/10/decolonising-artificial-intelligence/>.
- [30] M. Dehghan, Z. Zhang, M. Siam, J. Jin, L. Petrich, and M. Jagersand, “Online object and task learning via human robot interaction,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 2132–2138.
- [31] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,” in *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, Ieee, 2009, pp. 248–255.
- [32] T.-T. Do, A. Nguyen, and I. Reid, “Affordancenet: An end-to-end deep learning approach for object affordance detection,” in *International Conference on Robotics and Automation (ICRA)*, 2018.
- [33] N. Dong and E. P. Xing, “Few-shot semantic segmentation with prototype learning,” in *British Machine Vision Conference (BMVC)*, vol. 3, 2018, p. 4.
- [34] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, “Flownet: Learning optical flow with convolutional networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2758–2766.
- [35] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” *arXiv preprint arXiv:1711.03938*, 2017.

- [36] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox, “Discriminative unsupervised feature learning with convolutional neural networks,” in *Advances in Neural Information Processing Systems*, 2014, pp. 766–774.
- [37] B. Drayer and T. Brox, “Object detection, tracking, and motion segmentation for object-level video segmentation,” *arXiv preprint arXiv:1608.03066*, 2016.
- [38] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” *arXiv preprint arXiv:1603.07285*, 2016.
- [39] A. Dutta, A. Gupta, and A. Zissermann, *VGG image annotator (VIA)*, <http://www.robots.ox.ac.uk/vgg/software/via/>, 2016.
- [40] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes challenge: A retrospective,” *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.
- [41] A. Faktor and M. Irani, “Video segmentation by non-local consensus voting,” in *British Machine Vision Conference (BMVC)*, vol. 2, 2014, p. 8.
- [42] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, and R. Klette, “STFCN: spatio-temporal FCN for semantic video segmentation,” *CoRR*, vol. abs/1608.05971, 2016. [Online]. Available: <http://arxiv.org/abs/1608.05971>.
- [43] C. Finn, P. Abbeel, and S. Levine, “Model-agnostic meta-learning for fast adaptation of deep networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, PMLR, 2017, pp. 1126–1135.
- [44] C. Finn, K. Xu, and S. Levine, “Probabilistic model-agnostic meta-learning,” in *Advances in Neural Information Processing Systems*, 2018, pp. 9516–9527.
- [45] R. Gadde, V. Jampani, and P. V. Gehler, “Semantic video cnns through representation warping,” *CoRR*, abs/1708.03088, 2017.
- [46] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.
- [47] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [48] C. Geng, S.-j. Huang, and S. Chen, “Recent advances in open set recognition: A survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

- [49] S. Gidaris and N. Komodakis, “Dynamic few-shot visual learning without forgetting,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.
- [50] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow, “Digging into self-supervised monocular depth estimation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3828–3838.
- [51] J. Goldberger, G. E. Hinton, S. T. Roweis, and R. R. Salakhutdinov, “Neighbourhood components analysis,” in *Advances in Neural Information Processing Systems*, 2005, pp. 513–520.
- [52] J. Gou, B. Yu, S. J. Maybank, and D. Tao, “Knowledge distillation: A survey,” *arXiv preprint arXiv:2006.05525*, 2020.
- [53] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, IEEE, vol. 2, 2006, pp. 1735–1742.
- [54] B. Hariharan and R. Girshick, “Low-shot visual recognition by shrinking and hallucinating features,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3018–3027.
- [55] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
- [56] K. He, G. Gkioxari, P. Dollár, and R. Girshick, “Mask r-cnn,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2961–2969.
- [57] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.
- [58] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, “High-speed tracking with kernelized correlation filters,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 583–596, 2015.
- [59] C. F. Hester and D. Casasent, “Multivariant technique for multiclass pattern recognition,” *Applied Optics*, vol. 19, no. 11, pp. 1758–1761, 1980.
- [60] G. Hinton, *Neural Networks for Machine Learning, Lecture Notes: overview of mini-batch gradient descent*, URL: [https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf).

- [61] R. Hou, J. Li, A. Bhargava, A. Raventos, V. Guizilini, C. Fang, J. Lynch, and A. Gaidon, “Real-time panoptic segmentation from dense detections,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8523–8532.
- [62] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” *arXiv preprint arXiv:1704.04861*, 2017.
- [63] T.-I. Hsieh, Y.-C. Lo, H.-T. Chen, and T.-L. Liu, “One-shot object detection with co-attention and co-excitation,” in *Advances in Neural Information Processing Systems*, 2019, pp. 2721–2730.
- [64] Y.-T. Hu, J.-B. Huang, and A. Schwing, “Maskrnn: Instance level video object segmentation,” in *Advances in Neural Information Processing Systems*, 2017, pp. 325–334.
- [65] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, “FlowNet 2.0: Evolution of optical flow estimation with deep networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2462–2470.
- [66] S. D. Jain, B. Xiong, and K. Grauman, “Fusionseg: Learning to combine motion and appearance for fully automatic segmentation of generic objects in videos,” *arXiv preprint arXiv:1701.05384*, 2017.
- [67] A. Kae, K. Sohn, H. Lee, and E. Learned-Miller, “Augmenting CRFs with Boltzmann machine shape priors for image labeling,” in *CVPR*, 2013.
- [68] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *International journal of computer vision*, vol. 1, no. 4, pp. 321–331, 1988.
- [69] A. Kendall, V. Badrinarayanan, and R. Cipolla, “Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding,” *arXiv preprint arXiv:1511.02680*, 2015.
- [70] J. Kenney, T. Buckley, and O. Brock, “Interactive segmentation for manipulation in unstructured environments,” in *IEEE International Conference on Robotics and Automation, 2009. ICRA '09.*, IEEE, 2009, pp. 1377–1382.
- [71] M. Keuper, B. Andres, and T. Brox, “Motion trajectory segmentation via minimum cost multicuts,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 3271–3279.
- [72] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, “Supervised contrastive learning,” *arXiv preprint arXiv:2004.11362*, 2020.

- [73] D. Kim, S. Woo, J.-Y. Lee, and I. S. Kweon, “Video panoptic segmentation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9859–9868.
- [74] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [75] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, “Panoptic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.
- [76] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *International Conference on Machine Learning (ICML) Deep Learning Workshop*, vol. 2, 2015.
- [77] Y. J. Koh and C. Kim, “Primary object segmentation in videos based on region augmentation and reduction,” pp. 7417–7425, 2017. DOI: 10.1109/CVPR.2017.784.
- [78] P. Krähenbühl and V. Koltun, “Efficient inference in fully connected crfs with gaussian edge potentials,” in *Advances in Neural Information Processing Systems*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Q. Weinberger, Eds., vol. 24, Curran Associates, Inc., 2011. [Online]. Available: <https://proceedings.neurips.cc/paper/2011/file/beda24c1e1b46055dff2c39c98fd6fc1-Paper.pdf>.
- [79] A. Kundu, K. M. Krishna, and J. Sivaswamy, “Moving object detection by multi-view geometric techniques from a single camera mounted robot,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems, 2009. IROS 2009.*, IEEE, 2009, pp. 4306–4312.
- [80] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr, “Associative hierarchical crfs for object class image segmentation,” in *2009 IEEE 12th International Conference on Computer Vision*, IEEE, 2009, pp. 739–746.
- [81] B. Lake, R. Salakhutdinov, J. Gross, and J. Tenenbaum, “One shot learning of simple visual concepts,” in *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 33, 2011.
- [82] K. F. Lee, *The real threat of artificial intelligence*, 2017. [Online]. Available: <https://www.nytimes.com/2017/06/24/opinion/sunday/artificial-intelligence-economic-inequality.html>.
- [83] S. Lee, S. Im, S. Lin, and I. S. Kweon, “Instance-wise depth and motion learning from monocular videos,” *arXiv preprint arXiv:1912.09351*, 2019.

- [84] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, “Video segmentation by tracking many figure-ground segments,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2192–2199.
- [85] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2117–2125.
- [86] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2980–2988.
- [87] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European Conference on Computer Vision*, Springer, 2014, pp. 740–755.
- [88] C. Liu *et al.*, “Beyond pixels: Exploring new representations and applications for motion analysis,” Ph.D. dissertation, Massachusetts Institute of Technology, 2009.
- [89] H. Liu, K. Simonyan, and Y. Yang, “Darts: Differentiable architecture search,” *arXiv preprint arXiv:1806.09055*, 2018.
- [90] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, “An intriguing failing of convolutional neural networks and the coordconv solution,” in *Advances in Neural Information Processing Systems*, 2018, pp. 9605–9616.
- [91] V. Lomonaco and D. Maltoni, “Core50: A new dataset and benchmark for continuous object recognition,” *arXiv preprint arXiv:1705.03550*, 2017.
- [92] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.
- [93] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.
- [94] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, “See more, know more: Unsupervised video object segmentation with co-attention siamese networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3623–3632.
- [95] J. Luiten, P. Voigtlaender, and B. Leibe, “Premvos: Proposal-generation, refinement and merging for video object segmentation,” in *Asian Conference on Computer Vision*, Springer, 2018, pp. 565–580.



- [96] L. v. d. Maaten and G. Hinton, “Visualizing data using t-SNE,” *Journal of Machine Learning Research*, vol. 9, no. Nov, pp. 2579–2605, 2008.
- [97] E. M. Markman, *Categorization and Naming in Children: Problems of Induction*. Mit Press, 1989.
- [98] N. Mayer, E. Ilg, P. Hausser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox, “A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4040–4048.
- [99] F. Milletari, N. Navab, and S.-A. Ahmadi, “V-net: Fully convolutional neural networks for volumetric medical image segmentation,” in *2016 fourth international conference on 3D vision (3DV)*, IEEE, 2016, pp. 565–571.
- [100] E. Mohamed, M. Ewaisha, M. Siam, H. Rashed, S. Yogamani, and A. El-Sallab, “Instancemotseg: Real-time instance motion segmentation for autonomous driving,” *arXiv preprint arXiv:2008.07008*, 2020.
- [101] Y. Movshovitz-Attias, A. Toshev, T. K. Leung, S. Ioffe, and S. Singh, “No fuss distance metric learning using proxies,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 360–368.
- [102] T. Munkhdalai and H. Yu, “Meta networks,” in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, JMLR. org, 2017, pp. 2554–2563.
- [103] Naoki, *Up-sampling with transposed convolution*, Nov. 2017. [Online]. Available: <https://naokishibuya.medium.com/up-sampling-with-transposed-convolution-9ae4f2df52d0%7D>.
- [104] G. Neuhold, T. Ollmann, S. Rota Bulò, and P. Kotschieder, “The mapillary vistas dataset for semantic understanding of street scenes,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4990–4999.
- [105] A. Nichol and J. Schulman, “Reptile: A scalable metalearning algorithm,” *arXiv preprint arXiv:1803.02999*, vol. 2, 2018.
- [106] D. Nilsson and C. Sminchisescu, “Semantic video segmentation by gated recurrent flow propagation,” *arXiv preprint arXiv:1612.08871*, 2016.
- [107] H. Noh, S. Hong, and B. Han, “Learning deconvolution network for semantic segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1520–1528.
- [108] P. Ochs, J. Malik, and T. Brox, “Segmentation of moving objects by long term video analysis,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014.

- [109] A. Ošep, P. Voigtlaender, M. Weber, J. Luiten, and B. Leibe, “4d generic video object proposals,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2020, pp. 10 031–10 037.
- [110] A. Papazoglou and V. Ferrari, “Fast object segmentation in unconstrained video,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1777–1784.
- [111] G. Pasquale, C. Ciliberto, F. Odone, L. Rosasco, and L. Natale, “Teaching icub to recognize objects using deep convolutional neural networks,” in *Machine Learning for Interactive Systems*, 2015, pp. 21–25.
- [112] G. Pasquale, C. Ciliberto, L. Rosasco, and L. Natale, “Object identification from few examples by improving the invariance of a deep convolutional neural network,” in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2016*, IEEE, 2016, pp. 4904–4911.
- [113] A. Paszke, A. Chaurasia, S. Kim, and E. Culurciello, “Enet: A deep neural network architecture for real-time semantic segmentation,” *arXiv preprint arXiv:1606.02147*, 2016.
- [114] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems*, 2019, pp. 8026–8037.
- [115] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in *Computer Vision and Pattern Recognition*, 2016.
- [116] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, “A benchmark dataset and evaluation methodology for video object segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 724–732.
- [117] S. Pirk, M. Khansari, Y. Bai, C. Lynch, and P. Sermanet, “Online object representations with contrastive learning,” *arXiv preprint arXiv:1906.04312*, 2019.
- [118] H. Pirsiavash and D. Ramanan, “Detecting activities of daily living in first-person camera views,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012*, IEEE, 2012, pp. 2847–2854.
- [119] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, “The 2017 davis challenge on video object segmentation,” *arXiv preprint arXiv:1704.00675*, 2017.

- [120] H. Qi, M. Brown, and D. G. Lowe, “Low-shot learning with imprinted weights,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5822–5830.
- [121] K. Rakelly, E. Shelhamer, T. Darrell, A. Efros, and S. Levine, “Conditional networks for few-shot semantic segmentation,” 2018.
- [122] H. Rashed, M. Ramzy, V. Vaquero, A. El Sallab, G. Sistu, and S. Yogamani, “Fusemodnet: Real-time camera and lidar based moving object detection for robust low-light autonomous driving,” in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct. 2019.
- [123] S. Ravi and H. Larochelle, “Optimization as a model for few-shot learning,” in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, OpenReview.net, 2017. [Online]. Available: <https://openreview.net/forum?id=rJY0-Kc11>.
- [124] —, “Optimization as a model for few-shot learning,” in *International Conference on Learning Representations (ICLR)*, 2017.
- [125] H. Raza, M. Ravanbakhsh, T. Klein, and M. Nabi, “Weakly supervised one shot segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 2019, pp. 0–0.
- [126] N. D. Reddy, P. Singhal, and K. M. Krishna, “Semantic motion segmentation using dense crf formulation,” in *Proceedings of the 2014 Indian Conference on Computer Vision Graphics and Image Processing*, ACM, 2014, p. 56.
- [127] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779–788.
- [128] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, “Meta-learning for semi-supervised few-shot classification,” *arXiv preprint arXiv:1803.00676*, 2018.
- [129] K. Robotics, *KUKA Innovation Award Challenge*, [https://www.youtube.com/watch?v=aLcw73dt\\_0o](https://www.youtube.com/watch?v=aLcw73dt_0o), 2018.
- [130] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, Springer, 2015, pp. 234–241.
- [131] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3234–3243.

- [132] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, “Meta-learning with latent embedding optimization,” *arXiv preprint arXiv:1807.05960*, 2018.
- [133] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, “Meta-learning with memory-augmented neural networks,” in *International Conference on Machine Learning (ICML)*, 2016, pp. 1842–1850.
- [134] M. Schwarz, H. Schulz, and S. Behnke, “Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features,” in *IEEE International Conference on Robotics and Automation (ICRA), 2015*, IEEE, 2015, pp. 1329–1335.
- [135] G. Seguin, P. Bojanowski, R. Lajugie, and I. Laptev, “Instance-level video segmentation from object tracks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3678–3687.
- [136] A. Shaban, S. Bansal, Z. Liu, I. Essa, and B. Boots, “One-shot learning for semantic segmentation,” *arXiv preprint arXiv:1709.03410*, 2017.
- [137] M. P. Shah, “Semantic segmentation architectures implemented in PyTorch,” <https://github.com/meetshah1995/pytorch-semseg>, 2017.
- [138] J. Shotton, M. Johnson, and R. Cipolla, “Semantic texton forests for image categorization and segmentation,” in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, 2008, pp. 1–8.
- [139] J. Shotton, J. Winn, C. Rother, and A. Criminisi, “Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation,” in *European Conference on Computer Vision*, Springer, 2006, pp. 1–15.
- [140] M. Siam, *Multi-Task Learning with Motion and Appearance*, <http://webdocs.cs.ualberta.ca/~mennatul/?p=99>, 2017.
- [141] M. Siam, S. Eikerdawy, M. Gamal, M. Abdel-Razek, M. Jagersand, and H. Zhang, “Real-time segmentation with appearance, motion and geometry,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 5793–5800.
- [142] M. Siam, M. Gamal, M. Abdel-Razek, S. Yogamani, and M. Jagersand, “Rtseg: Real-time semantic segmentation comparative study,” *arXiv preprint arXiv:1803.02758*, 2018.
- [143] M. Siam, C. Jiang, S. Lu, L. Petrich, M. Gamal, M. Elhoseiny, and M. Jagersand, “Video object segmentation using teacher-student adaptation in a human robot interaction (hri) setting,” in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 50–56.

- [144] M. Siam, A. Kendall, and M. Jagersand, “Video class agnostic segmentation benchmark for autonomous driving,” *arXiv preprint arXiv:2103.11015*, 2021.
- [145] ———, “Video class agnostic segmentation with contrastive learning in autonomous driving,” in *Under Review for IEEE International Conference on Computer Vision.*, 2021.
- [146] M. Siam, H. Mahgoub, M. Zahran, S. Yogamani, M. Jagersand, and A. El-Sallab, “Modnet: Moving object detection network with motion and appearance for autonomous driving,” *arXiv preprint arXiv:1709.04821*, 2017.
- [147] M. Siam, B. N. Oreshkin, and M. Jagersand, “Amp: Adaptive masked proxies for few-shot segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5249–5258.
- [148] M. Siam, S. Yogamani, A. ElSallab, and H. Mahgoub, *Moving object detection network*, Aug. 2019. [Online]. Available: [https://worldwide.espacenet.com/publicationDetails/biblio?CC=DE&NR=102018114229&KC=&FT=E&locale=en\\_EP#%7D](https://worldwide.espacenet.com/publicationDetails/biblio?CC=DE&NR=102018114229&KC=&FT=E&locale=en_EP#%7D).
- [149] C. Simon, P. Koniusz, R. Nock, and M. Harandi, “Adaptive subspaces for few-shot learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 4136–4145.
- [150] K. Simonyan and A. Zisserman, “Two-stream convolutional networks for action recognition in videos,” in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.
- [151] ———, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [152] A. Singh, J. Sha, K. S. Narayan, T. Achim, and P. Abbeel, “Bigbird: A large-scale 3d database of object instances,” in *IEEE International Conference on Robotics and Automation (ICRA), 2014*, IEEE, 2014, pp. 509–516.
- [153] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.
- [154] H. Song, W. Wang, S. Zhao, J. Shen, and K.-M. Lam, “Pyramid dilated deeper convlstm for video salient object detection,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 715–731.
- [155] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Proceedings. 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Cat. No PR00149)*, IEEE, vol. 2, 1999, pp. 246–252.

- [156] J. Steward, D. Lichti, J. Chow, R. Ferber, and S. Osis, “Performance assessment and calibration of the kinect 2.0 time-of-flight range camera for use in motion capture applications,” in *Proceedings of the Fig Working Week*, 2015.
- [157] D. G. Stork, R. O. Duda, P. E. Hart, and D. Stork, “Pattern classification,” *A Wiley-Interscience Publication*, 2001.
- [158] P. Sturgess, K. Alahari, L. Ladicky, and P. H. Torr, “Combining appearance and structure from motion features for road scene understanding,” in *British Machine Vision Conference (BMVC)*, BMVA, 2009.
- [159] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, “Meta-transfer learning for few-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.
- [160] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, “Learning to compare: Relation network for few-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.
- [161] B. Taylor, V. Karasev, and S. Soatto, “Causal video object segmentation from persistence of occlusions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4268–4276.
- [162] M. Teichmann, M. Weber, M. Zoellner, R. Cipolla, and R. Urtasun, “Multinet: Real-time joint semantic reasoning for autonomous driving,” *arXiv preprint arXiv:1612.07695*, 2016.
- [163] Y. Tian, D. Krishnan, and P. Isola, “Contrastive multiview coding,” *arXiv preprint arXiv:1906.05849*, 2019.
- [164] Z. Tian, X. Lai, L. Jiang, M. Shu, H. Zhao, and J. Jia, “Generalized few-shot semantic segmentation,” *arXiv preprint arXiv:2010.05210*, 2020.
- [165] P. Tokmakov, K. Alahari, and C. Schmid, “Learning motion patterns in videos,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017, pp. 531–539.
- [166] —, “Learning video object segmentation with visual memory,” *arXiv preprint arXiv:1704.05737*, 2017.
- [167] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Deep end2end voxel2voxel prediction,” in *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2016*, IEEE, 2016, pp. 402–409.
- [168] E. Triantafillou, T. Zhu, V. Dumoulin, P. Lamblin, K. Xu, R. Goroshin, C. Gelada, K. Swersky, P.-A. Manzagol, and H. Larochelle, “Meta-dataset: A dataset of datasets for learning to learn from few examples,” *arXiv preprint arXiv:1903.03096*, 2019.

- [169] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, R. Caruana, A. Mohamed, M. Philipose, and M. Richardson, “Do deep convolutional nets really need to be deep and convolutional?” *arXiv preprint arXiv:1603.05691*, 2016.
- [170] W. R. Uttal, L. Spillmann, F. Stürzel, and A. B. Sekuler, “Motion and shape in common fate,” *Vision Research*, vol. 40, no. 3, pp. 301–310, 2000.
- [171] G. Varma, A. Subramanian, A. Namboodiri, M. Chandraker, and C. Jawahar, “Idd: A dataset for exploring problems of autonomous navigation in unconstrained environments,” in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, IEEE, 2019, pp. 1743–1751.
- [172] J. Vertens, A. Valada, and W. Burgard, “Smsnet: Semantic motion segmentation using deep convolutional neural networks,” in *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017.
- [173] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, *et al.*, “Matching networks for one shot learning,” in *Advances in Neural Information Processing Systems*, 2016, pp. 3630–3638.
- [174] P. Voigtlaender, Y. Chai, F. Schroff, H. Adam, B. Leibe, and L.-C. Chen, “Feelvos: Fast end-to-end embedding learning for video object segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9481–9490.
- [175] P. Voigtlaender, M. Krause, A. Osep, J. Luiten, B. B. G. Sekar, A. Geiger, and B. Leibe, “Mots: Multi-object tracking and segmentation,” in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
- [176] P. Voigtlaender and B. Leibe, “Online adaptation of convolutional neural networks for video object segmentation,” *arXiv preprint arXiv:1706.09364*, 2017.
- [177] K. Wang, J. H. Liew, Y. Zou, D. Zhou, and J. Feng, “Panet: Few-shot image semantic segmentation with prototype alignment,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9197–9206.
- [178] W. Wang, H. Song, S. Zhao, J. Shen, S. Zhao, S. C. Hoi, and H. Ling, “Learning unsupervised video object segmentation through visual attention,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3064–3074.
- [179] X. Wang, T. Kong, C. Shen, Y. Jiang, and L. Li, “Solo: Segmenting objects by locations,” *arXiv preprint arXiv:1912.04488*, 2019.

- [180] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, “Low-shot learning from imaginary data,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7278–7286.
- [181] Y. Wang and Q. Yao, “Few-shot learning: A survey,” *CoRR*, vol. abs/1904.05046, 2019. arXiv:1904.05046. [Online]. Available: <http://arxiv.org/abs/1904.05046>.
- [182] Z. Wang, J. Xu, L. Liu, F. Zhu, and L. Shao, “Ranet: Ranking attention network for fast video object segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3978–3987.
- [183] S. Wehrwein and R. Szeliski, “Video segmentation with background motion models,” in *British Machine Vision Conference (BMVC)*, 2017.
- [184] T. Wei, X. Li, Y. P. Chen, Y.-W. Tai, and C.-K. Tang, “Fss-1000: A 1000-class dataset for few-shot segmentation,” *arXiv preprint arXiv:1907.12347*, 2019.
- [185] J. Winkens, R. Bunel, A. G. Roy, R. Stanforth, V. Natarajan, J. R. Ledsam, P. MacWilliams, P. Kohli, A. Karthikesalingam, S. Kohl, *et al.*, “Contrastive training for improved out-of-distribution detection,” *arXiv preprint arXiv:2007.05566*, 2020.
- [186] K. Wong, S. Wang, M. Ren, M. Liang, and R. Urtasun, “Identifying unknown instances for autonomous driving,” in *Conference on Robot Learning*, PMLR, 2020, pp. 384–393.
- [187] Y. Wu and K. He, “Group normalization,” in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 3–19.
- [188] Z. Wu, A. A. Efros, and S. X. Yu, “Improving generalization via scalable neighborhood component analysis,” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 685–701.
- [189] Z. Wu, Y. Xiong, S. X. Yu, and D. Lin, “Unsupervised feature learning via non-parametric instance discrimination,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3733–3742.
- [190] Z. Wu, C. Shen, and A. v. d. Hengel, “Bridging category-level and instance-level semantic image segmentation,” *arXiv preprint arXiv:1605.06885*, 2016.
- [191] —, “Wider or deeper: Revisiting the resnet model for visual recognition,” *arXiv preprint arXiv:1611.10080*, 2016.
- [192] Y. Xian, S. Choudhury, Y. He, B. Schiele, and Z. Akata, “Semantic projection network for zero-and few-label semantic segmentation,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8256–8265.



- [193] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, “Upsnet: A unified panoptic segmentation network,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8818–8826.
- [194] N. Xu, L. Yang, Y. Fan, D. Yue, Y. Liang, J. Yang, and T. Huang, “Youtube-vos: A large-scale video object segmentation benchmark,” *arXiv preprint arXiv:1809.03327*, 2018.
- [195] L. Yang, Y. Fan, and N. Xu, “Video instance segmentation,” *arXiv preprint arXiv:1905.04804*, 2019.
- [196] Z. Yang, Q. Wang, L. Bertinetto, W. Hu, S. Bai, and P. H. Torr, “Anchor diffusion for unsupervised video object segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 931–940.
- [197] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, “Stacked attention networks for image question answering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 21–29.
- [198] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Computing Surveys (CSUR)*, vol. 38, no. 4, 13–es, 2006.
- [199] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.
- [200] C. Zhang, G. Lin, F. Liu, J. Guo, Q. Wu, and R. Yao, “Pyramid graph networks with connection attentions for region-based one-shot semantic segmentation,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 9587–9595.
- [201] C. Zhang, G. Lin, F. Liu, R. Yao, and C. Shen, “Canet: Class-agnostic segmentation networks with iterative refinement and attentive few-shot learning,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5217–5226.
- [202] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” *arXiv preprint arXiv:1707.01083*, 2017.
- [203] X. Zhang, Y. Wei, Y. Yang, and T. Huang, “SG-One: Similarity guidance network for one-shot semantic segmentation,” *arXiv preprint arXiv:1810.09091*, 2018.
- [204] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, “Conditional random fields as recurrent neural networks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 1529–1537.

- [205] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, “Unsupervised learning of depth and ego-motion from video,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1851–1858.